Présentation des performances paralléles du code QDD CUDA Fortran sur différentes Architectures GPU

Alejandro Estaña IR CNRS (CALMIP)

Mai Dinh
Jordan Héraud
Marc Vincendon
Eric Suraud

The real-time TDDFT code “Quantum Dissipative Dynamics” on a GPU. *Accepted. Computer Physics Communications*

- Engineering process
- Speed-up results
- Conclusion

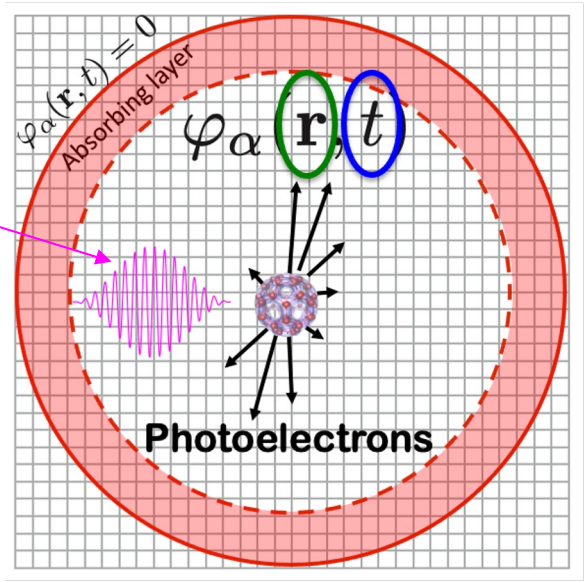
QDD Quantum Dissipative Dynamics

Study incoherent dynamical correlations in clusters and molecules

correlations beyond mean field dynamics

dominant in far off equilibrium situations

large excitation energies



Time-Dependent Density Functional Theory (TDDFT)
for (valence) electrons

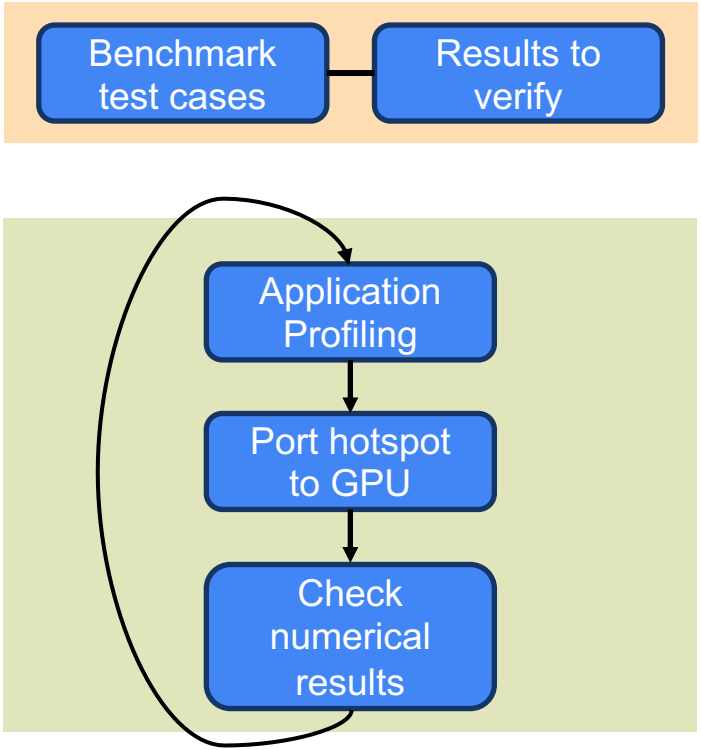
1) **real-time** propagation (not linear response)

$\varphi_\alpha(\mathbf{r},t) \overset{\text{TDDFT}}{\longrightarrow} \varphi_\alpha(\mathbf{r},t+\mathrm{d}t)$ by solving $\mathrm{i}\partial_t\varphi_\alpha = \widehat{h}_{\text{KS}}\,\varphi_\alpha$

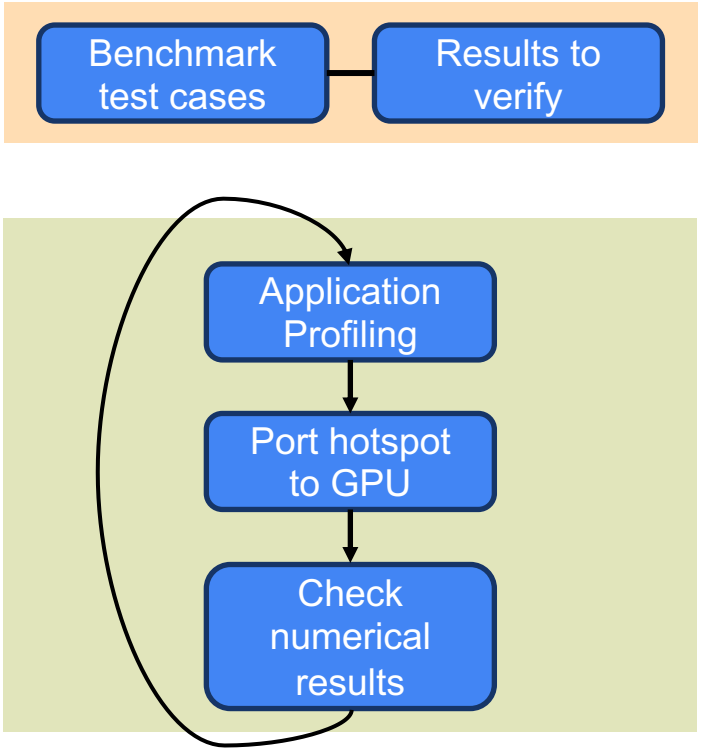
2) **real-space** description (no expansion on a basis of w.f.)
Discretization of $\varphi_\alpha(\mathbf{r},t)$ on 3D cartesian grid

Wopperer, Dinh, Reinhard, Suraud, Phys. Rep. 562 (2015) 1

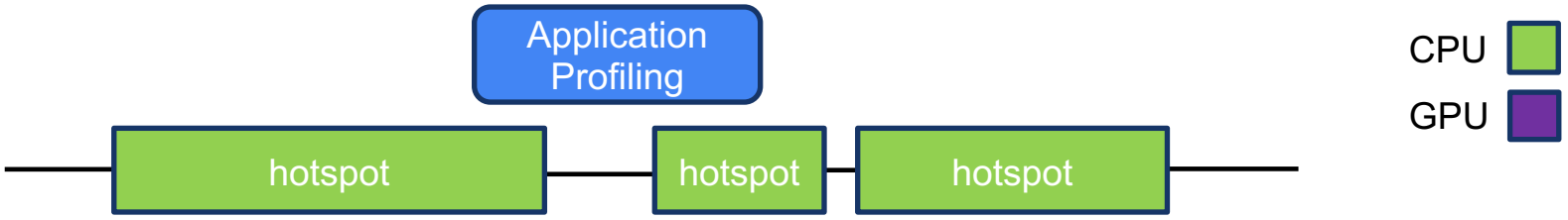
Engineering process



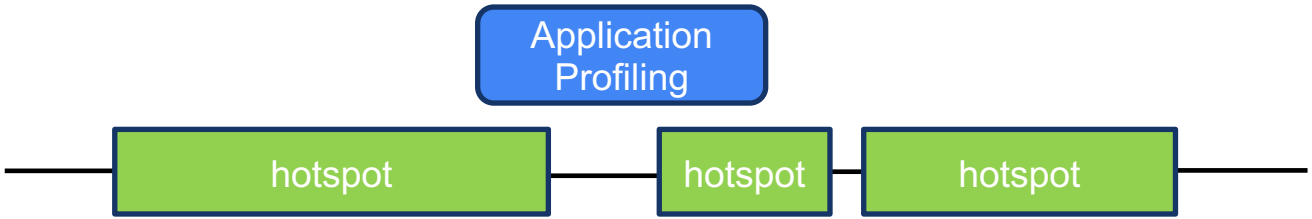
Engineering process



Engineering process



Engineering process

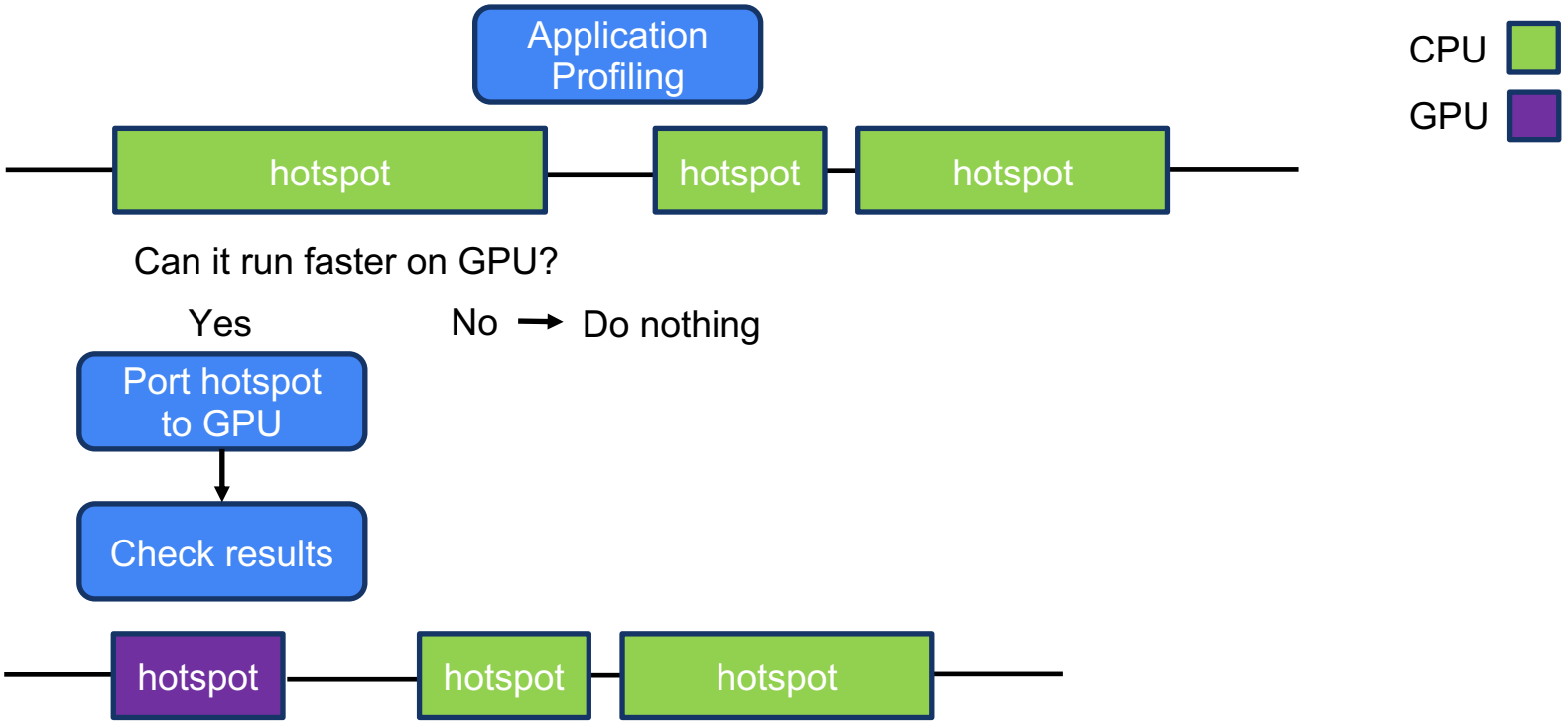


Can it run faster on GPU?

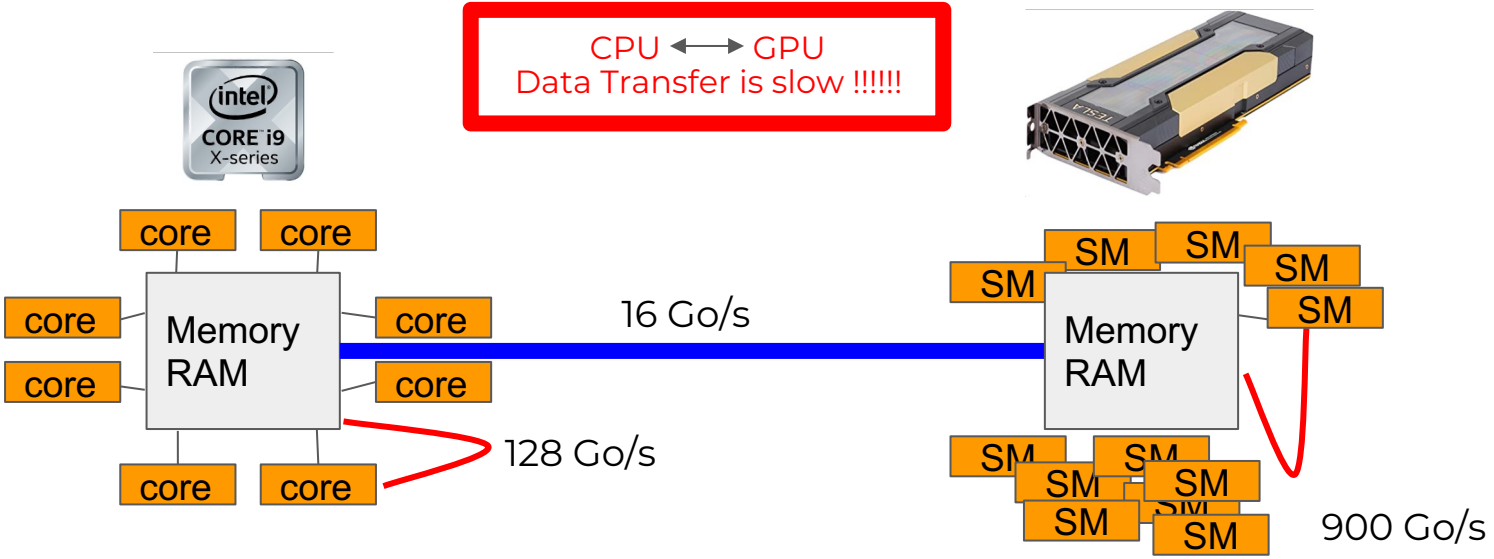
Yes

No → Do nothing

Engineering process



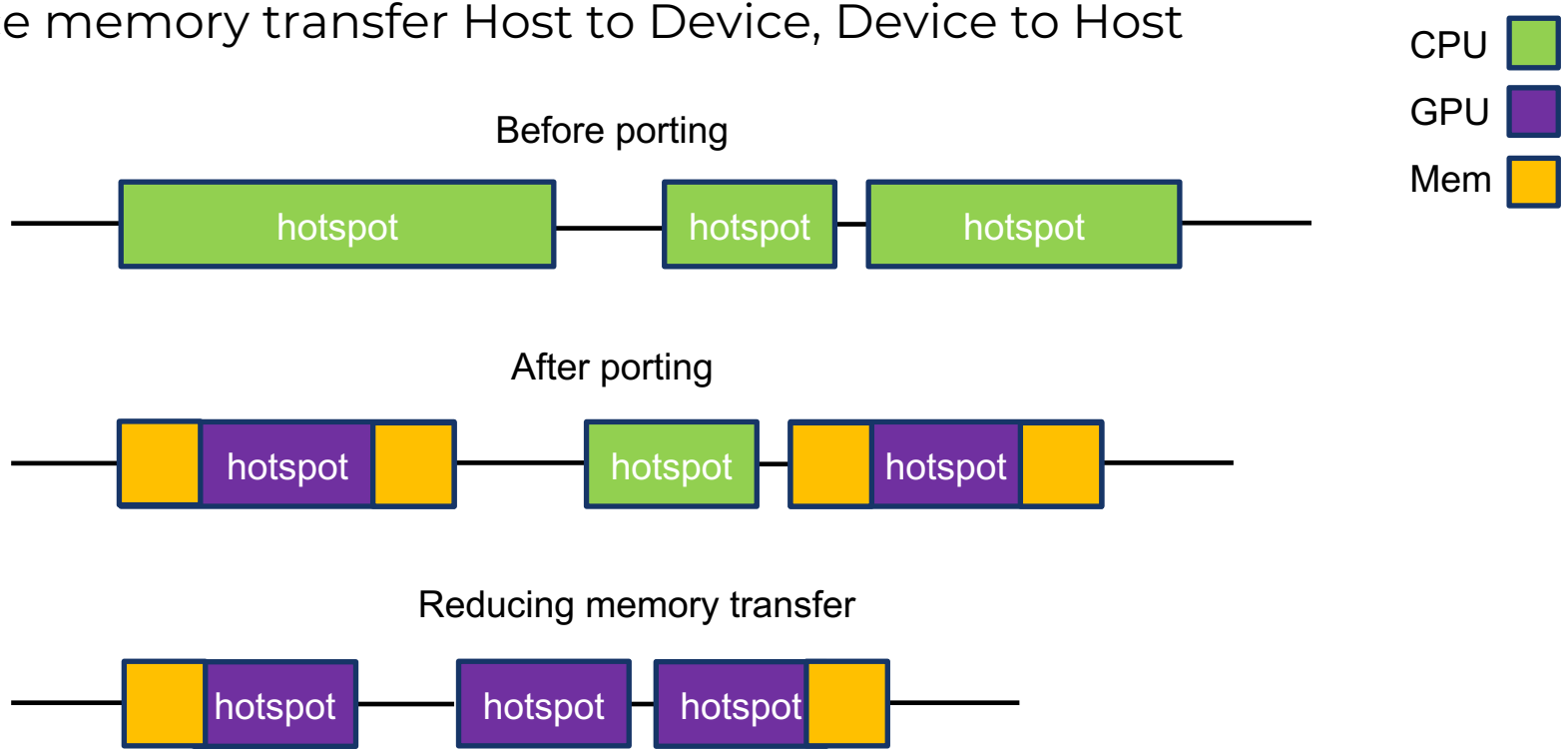
GPU Computing



Nvidia A100: 108 SM with 64 CUDA cores each

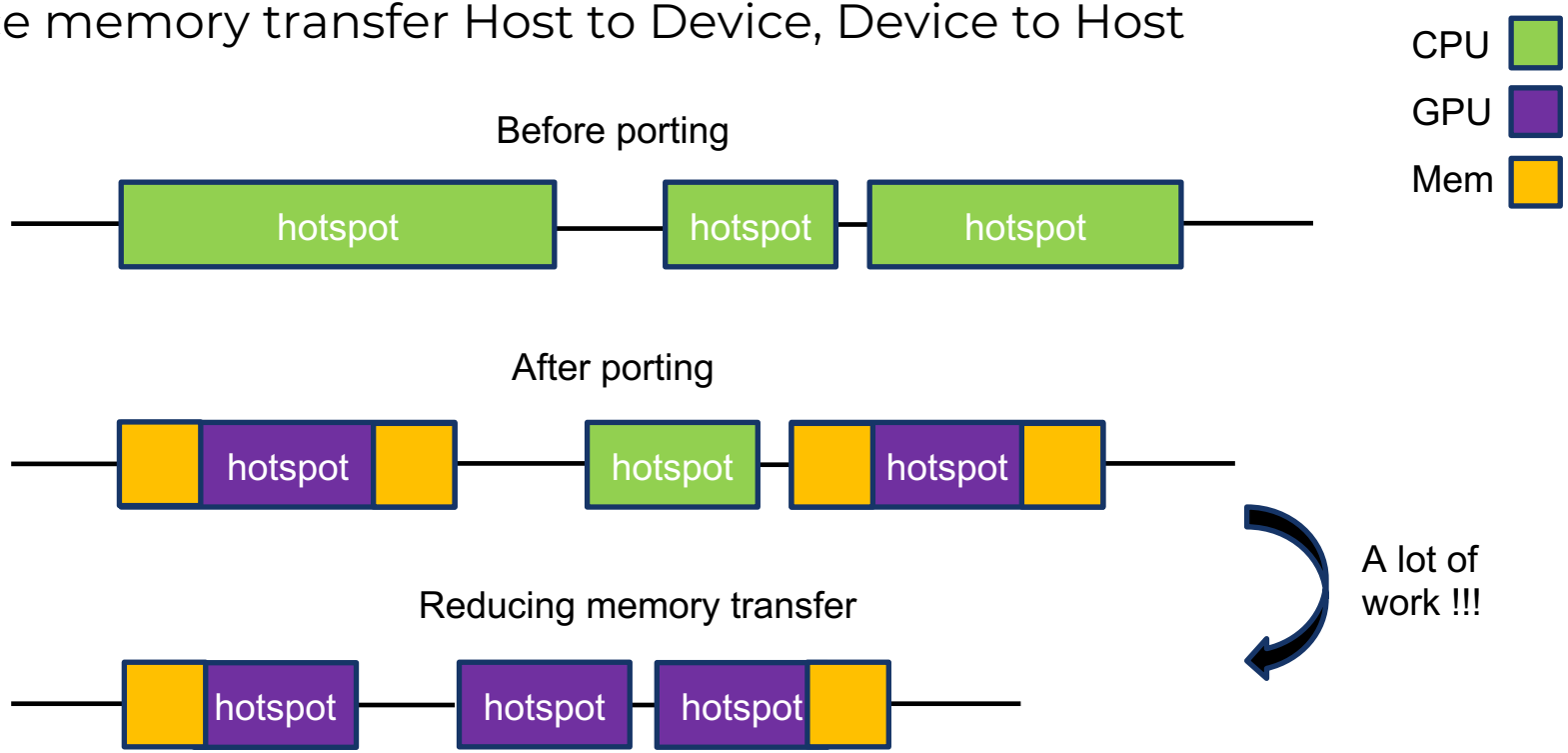
Engineering process

Reduce memory transfer Host to Device, Device to Host

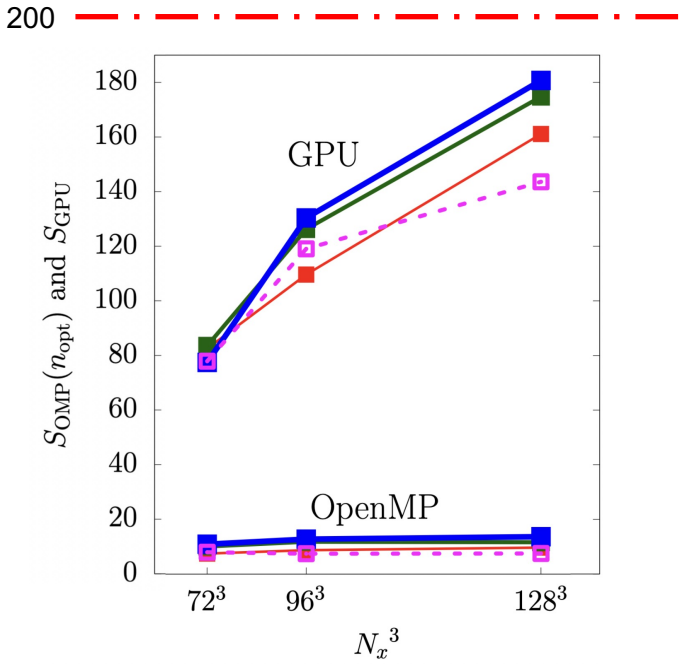


Engineering process

Reduce memory transfer Host to Device, Device to Host



Speed-up results

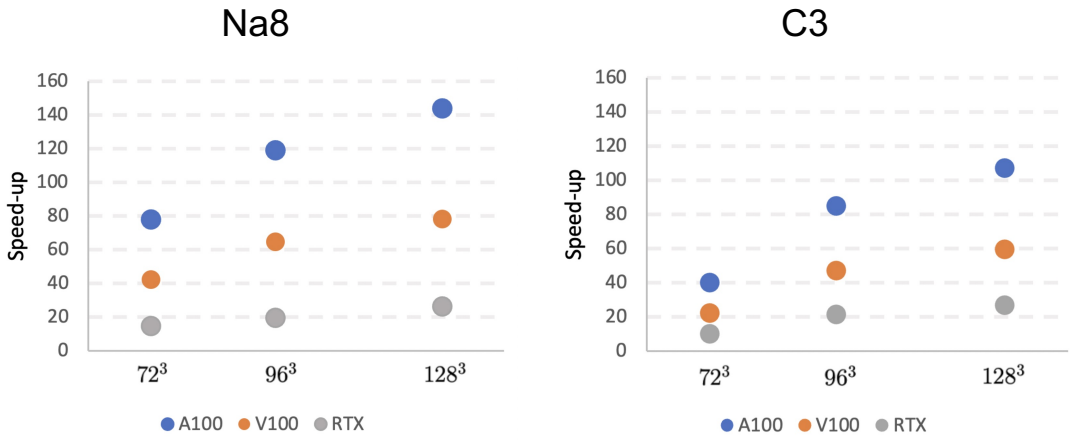


Nvidia A100

AMD epyc 7452

The real-time TDDFT code “Quantum Dissipative Dynamics” on a GPU. *Accepted. Computer Physics Communications*

Speed-up

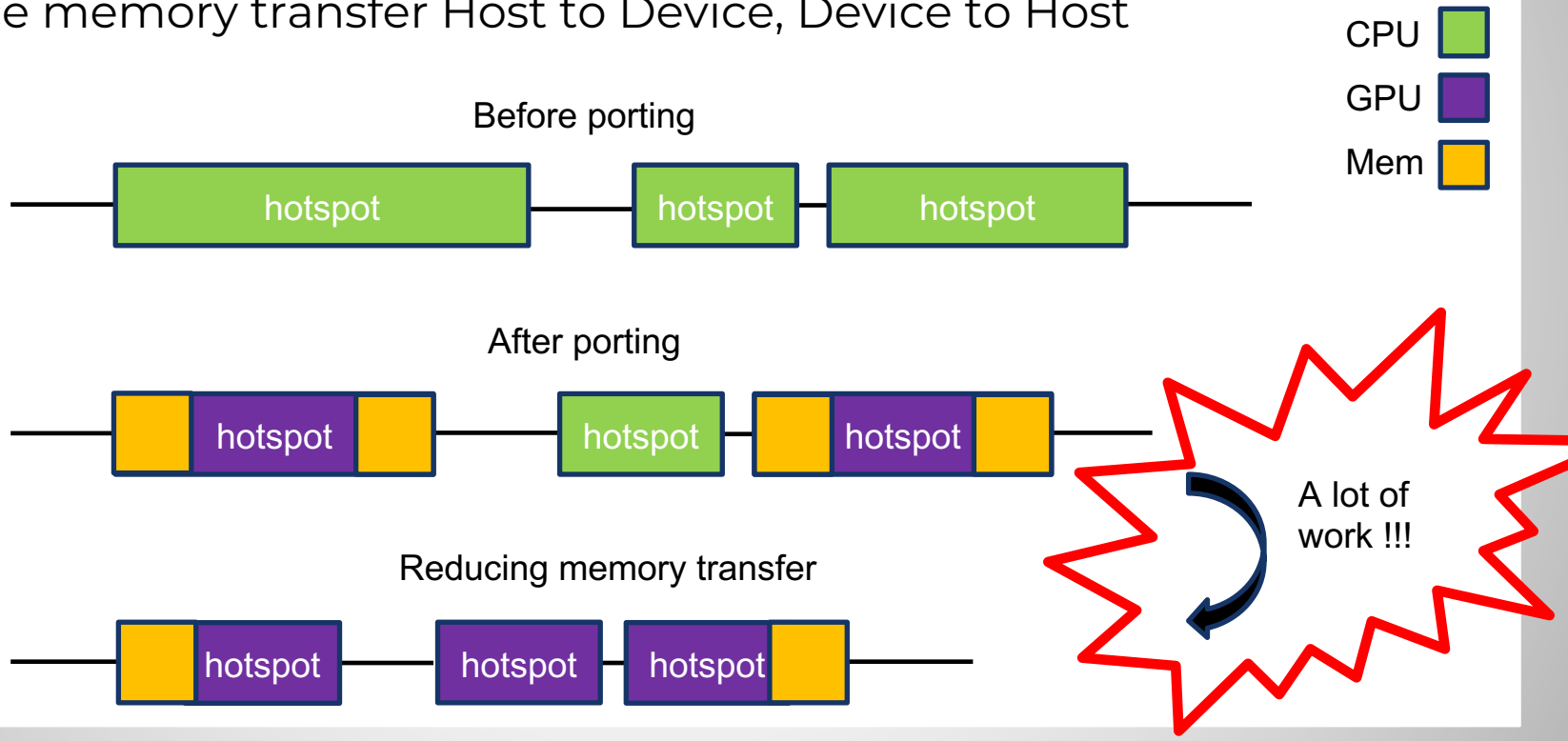


GPUs bandwith

Nvidia A100	1555	GB/s
Nvidia V100	900	GB/s
Nvidia RTX	416	GB/s

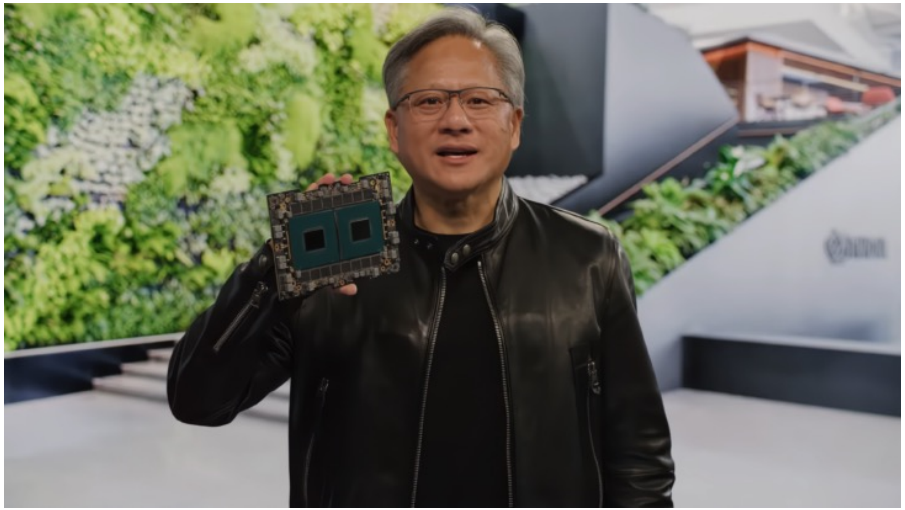
Performance Engineering process

Reduce memory transfer Host to Device, Device to Host



New CPU-GPU generation

Nvidia Grace Hopper



AMD APU MI300a

Conclusion

- The results obtained will allow exploring new methods in physics
- To achieve good performance:
  - Porting functions to GPU
  - Reducing CPU-GPU communication. Already deprecated ???
- Looking forward to testing new GPU-CPU architectures

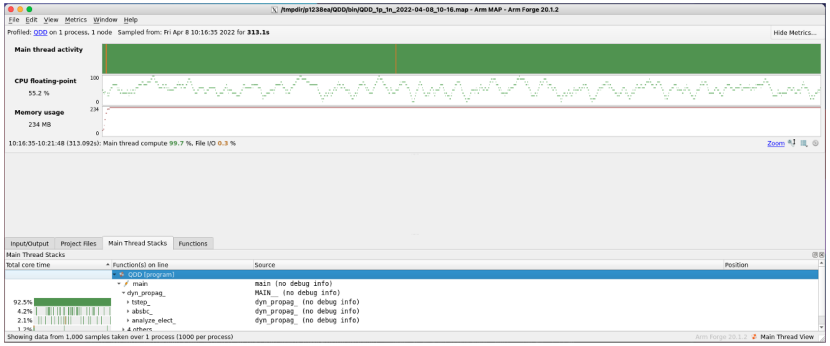
Thank you for your attention

Alejandro Estaña IR CNRS (CALMIP)

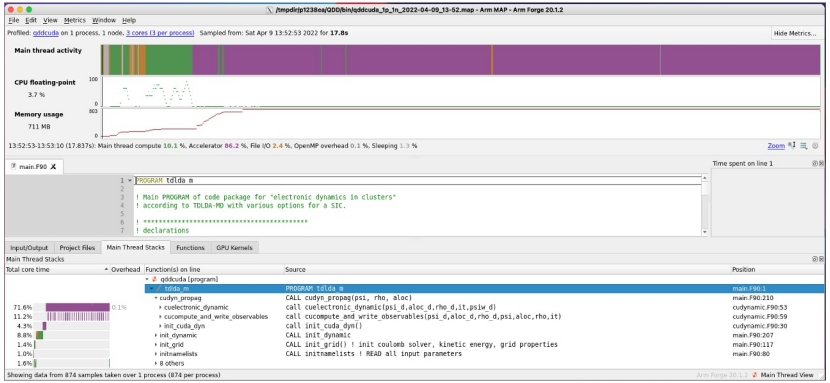
Mai Dinh
Jordan Héraud
Marc Vincendon
Eric Suraud

Arm Forge Map Profiling

Before

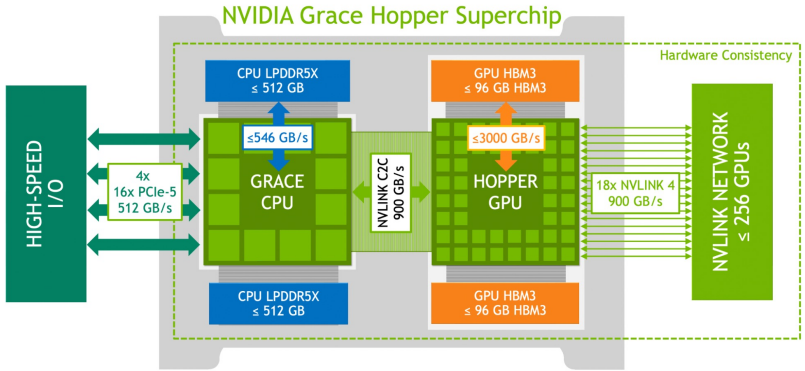


After



New CPU-GPU generation

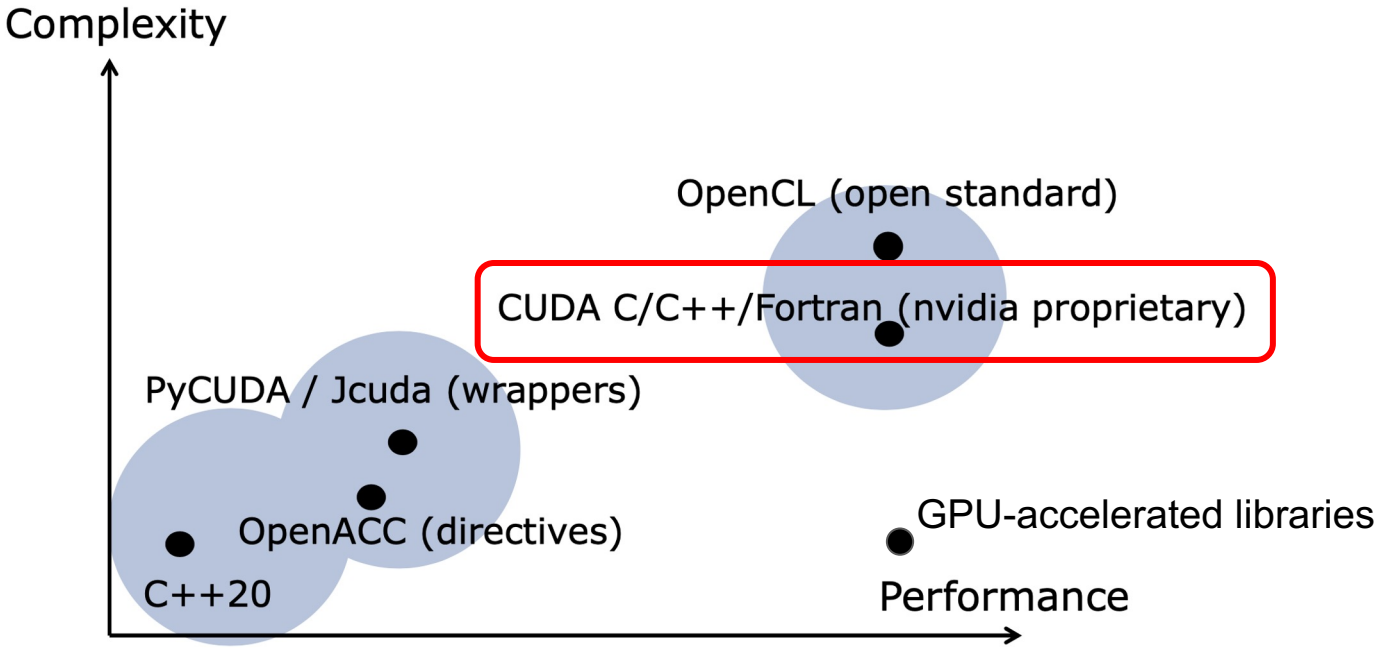
Nvidia Grace Hopper



AMD APU MI300a

QDD using CUDA language

Programming languages



Fortran to CUDA example

Fortran

Call fonction

```
CALL addfunctofield(field, func, x0, y0, z0, fact, para)
```

CUDA kernel

Call fonction

```
CALL cuaddfunctofield<<<dim3(ceiling(maxx/8), ceiling(maxy/8), ceiling(maxz/8)),dim3(8,8,4)>>>&
                      (field_d, func, x0, y0, z0, fact, para)
```

Fortran to CUDA example

Fortran

Call fonction

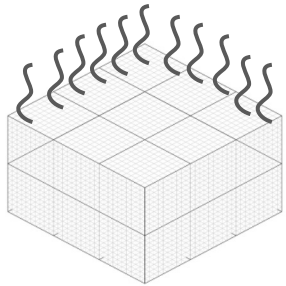
```
CALL addfunctofield(field, func, x0, y0, z0, fact, para)
```

CUDA kernel

Call fonction

```
CALL cuaddfunctofield<<<dim3(ceiling(maxx/8), ceiling(maxy/8), ceiling(maxz/8)),dim3(8,8,4)>>>&
                     (field_d, func, x0, y0, z0, fact, para)
```

```
CALL cuaddfunctofield<<<dim3(ceiling(maxx/8), ceiling(maxy/8), ceiling(maxz/8)),dim3(8,8,4)>>>&
                     (field_d, func, x0, y0, z0, fact, para)
```

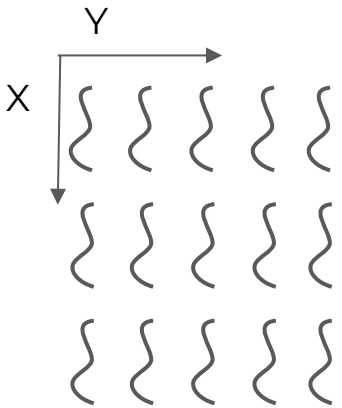


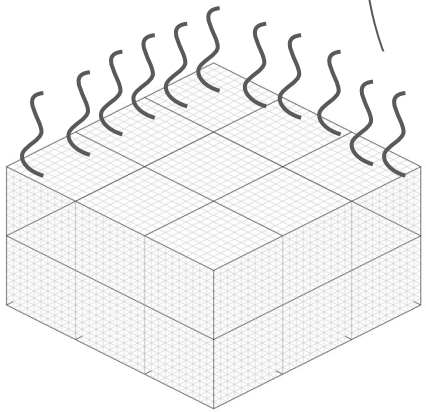
Fortran to CUDA example

CUDA kernel

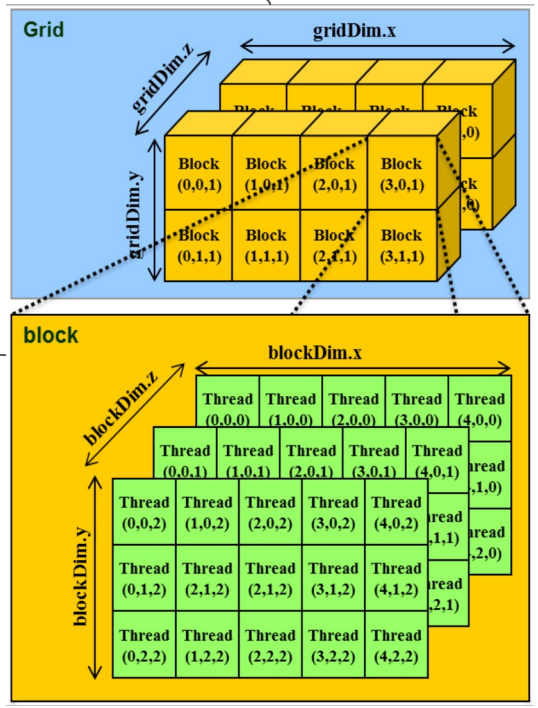
Call fonction

```
CALL cuaddfunctofield<<<dim3(ceiling(maxx/8), ceiling(maxy/8), ceiling(maxz/4)),dim3(8,8,4)>>>&
                                (field_d, func, x0, y0, z0, fact, para)
```





65.536 threads in parallel



Extra: GPU vs OpenMP