

Hackathon log book - LMDZ Team

doodle debriefing interne :

<https://doodle.com/poll/fgh4xr9dpkfy4ii>

doodle debriefing large :

<https://doodle.com/poll/68rv2zp65e6rsew3>

Debriefing interne Jeudi 10 juin

- intérêt d'un rapport ? pas trop
- plus utile : note de méthodologie
- date debriefing large :
Mer 23/06 14:00 29/06 14:00
- reste-t-il du travail sur la physique simplifiée
 - sorties : transférer sur CPU ou pas ?
 - tiling ?
- contenu debriefing
 - récit : Ehouarn, Laurent, Anne, Romain
 - outils : driver autonome, tests non-régression, nsight systems, nsight compute
 - OpenACC / note méthodo
 - code : exemples
 - résultats
 - discussion
 - lucidité sur espoirs en termes de perf non-parallélisme des parties CPU
 - par quoi commencer sur LMDZ ? et avec qui ?
 - instrumenter (dump) ?
 - équivalent du driver ?
 - turbulence / bucket
 - statut des dvp vis-à-vis du trunk LMDZ
 - ce qui peut en ressortir
 - nouveau rdv à l'automne pour attaquer une partie de LMDZ
 - consensus provisoire sur statut des dvp vis-à-vis trunk LMDZ

Day 4 Tuesday May 25

- final final refactoring of convection => Ehouarn, Anne, Sébastien => slightly worse
- radiative_lw => Dmitry
- scaling table / graph => Thomas,
<https://docs.google.com/spreadsheets/d/1XxjlvzEUIg19BAXyKE7tu5KCUqO16TxdNqVI7SvSZKs/edit>
- fake parallel CPU run on the driver =>

- slides =>
 - 1 : see Day 1
 - 2 : see Day 1
 - 3 : see Day 1
 - 4 : achievements
 - 5 : what did you learn
 - 6 : wishlist : tools / standards / event / systems
 - 7 : was it worth it ? ext steps, future

Day 3 Tuesday May 25

Nsight compute

module load nvidia-nsight-compute

- Find out which metrics you can look at:

```
LC_ALL=C ncu --metrics dram__bytes.sum.per_second,gpu__time_duration.avg --csv
--log-file lmdz-profile.csv ../driver/driver_gpu.exe
```

```
ncu --query-metrics | less
dram__bytes_read.sum
dram__bytes_read.sum.per_second
dram_bytes_write.sum
dram_bytes_write.sum.per_second
```

- Each metric needs a suffix (in *italics*, it is not shown by the previous command). To check for the suffix:


```
ncu --query-metrics --query-metric-mode suffix --metrics gpu__time_duration,
<other_metrics>
```
- To profile the code:


```
ncu --metrics dram__bytes_read.sum,<other_metrics.with_suffix>
./application
```

This will produce a small text report for each kernel

--print-units base option will use nanoseconds and bytes as units

--csv option will output more machine-readable format that you can parse

- First look at bandwidth : how much bandwidth is needed, how much is achieved

Async

- \$acc kernels async peut éliminer de l'attente de petits kernels
- ne pas mélanger async sync dans la même routine
- !\$acc wait en fin de routine : sinon la routine suivante qui n'utilise pas async peut commencer à calculer avant que notre calcul async soit fini

\$acc data*

- une unique region data
- kernel default(none)
- data create
 - tableaux locaux : create
 - tableaux en argument : copyin / copyout / copy
 - tableaux en module : copyin / copyout / copy

\$acc declare / \$acc create

- \$acc declare sur toutes les variables de module ALLOCATABLE

Day 2 Tuesday May 25

Zoom with mentors on May 20 3pm

<https://ecolepolytechnique.zoom.us/j/86525067303>

Suggestions

- produce OpenACC code for cpu with `acc=multicore` or both with `acc=XXX,multicore`

Discussion

- data management in called subroutine
 - open/close structured data section
`!$acc data ... !$acc end data`
 - create/delete local arrays
`!$acc data create(...)`
 - declare array arguments as `present / copyin / copyout`
 - NOTE : `copyin/copyout` **does nothing if array is already on the GPU**
use `update` to force copy back to CPU (e.g. before a section not yet ported to OpenACC)
 - use clause `default(none)` in loops/parallel regions to trigger error if we forgot to declare an array
 - or use clause `default(present)`
 - **rationale : in addition to allocation/copy at runtime, these directives let the compiler know about the status of arrays and avoid unnecessary transfers**
- data management in calling subroutine
 - open/close structured data section with clauses for arrays passed as arguments in CALLs
 - in the called subroutine, the corresponding arguments should be declared `present` or `copyin`
- convection/convadj :

- declare subroutine onecolumn as `seq` with `!$acc routine`
 - call subroutine from parallel loop
- if time allows : in `soil_mod` we have two-level nested loops where
 - the inner loop (horizontal index) is vectorizable
 - the outer loop (vertical index) has dependencies

so `collapse(2)` is incorrect, but surely there is better way than just having `!$acc parallel loop` before the inner loop.

This pattern is similar to lines 104-109 of

https://gitlab.in2p3.fr/ipsl/projets/dynamico/dynamico/-/blob/trunk2master/src/dynamics/caldyn_kernels_base.F90 and to lines 250-258 of

https://gitlab.in2p3.fr/ipsl/projets/dynamico/dynamico/-/blob/trunk2master/src/dynamics/caldyn_kernels_base.F90

Between May 17 and May 25

Memory management

- `radiative_sw`
- `radiative_lw` Anne, Sébastien, Laurent
- `turbulence (vdif)`
- `convection` Thomas
- `soil_mod` Laurent, Thomas

Porting to GPU (kernels)

- `radiative_sw` Thomas
- `radiative_lw` Anne, Laurent, Sébastien
- `turbulence (vdif)` Eliott, Ehouarn
- `radiative_mod` Romain
- `convection` Thomas
- `soil_mod :` Laurent, Sébastien
- `phyparam` Ehouarn
- `astronomy`
- `orbite, solarlong`
- `solar`
- `solang, zenang, mucorr`

Day 1 Monday May 17

Autocompare

- compile with `-ta=tesla:cc70,autocompare`
- autocompare verifies the full array even if computation is on part of the array. In this case it may be required to initialize the full array to something

- CPU and GPU computations can differ due to round-off error or different implementation of transcendental functions. A tolerance can be set with the environment variable PCAST_COMPARE
<https://developer.nvidia.com/blog/detecting-divergence-using-pcast-to-compare-gpu-to-cpu-results/>
<https://docs.nvidia.com/hpc-sdk/compiler/hpc-compilers-user-guide/#pcast-env-vars>
- see phyparam/bash/hackathon.sh

OpenACC

- first pass (radiative_sw.F90): just surround compute region with
!\$acc kernels
 ...
!\$acc end kernels
- more advanced (radiative_lw.F90)
 - surround compute region with enter/exit clauses + data clauses
!\$acc enter data create(zup(:,,:), zplanck(:,,:), ...)
 ...
!\$acc exit data delete(zup(:,,:), zplanck(:,,:), ...)
 - inside the compute region, prepend innermost loops with
!\$acc parallel loop

Profiling

- on compute nodes, make sure to set TMPDIR and link /tmp/nvidia to it
export TMPDIR=\$JOBSCRATCH
ln -s \$JOBSCRATCH /tmp/nvidia
- srun with appropriate options
srun --nsys profile -t openacc -b dwarf -o profile_%q{SLURM_PROCID}
../modeles/DYNAMICO_phyparam/bin/dynamico_phyparam.exe >> gcm.log
- visualize *.qdrep with nsys-ui :
nsys-ui profile_0.qdrep &
 preferably on local machine. If needed, register with Nvidia and download Nsight Systems :
<https://developer.nvidia.com/gameworksdownload#?dn=nsight-systems-2021-2-1-58>
 On ubuntu, the DEB package can be installed with **sudo apt-get install** .

Meeting with mentors Tuesday May 4

- <https://tinyurl.com/4z2pw8nz>
- agenda : <https://drive.google.com/file/d/1tIOX7Eyz-vIMJWYqjQ9rBzU5ICGksDH-/view>
- penser / préparer aux visuels à présenter => perf ?
- profiler

Pre-Hackathon

Ressources

git :

- https://gitlab.in2p3.fr/ipsi/projets/dynamico/simple_physics
- <https://github.com/progit/progit2/releases/download/2.1.303/progit.pdf>

OpenACC :

- IDRIS : http://www.idris.fr/media/formations/openacc/gpu_directives.pdf
- NVidia :
- autocompare :
<https://developer.nvidia.com/blog/detecting-divergence-using-pcast-to-compare-gpu-to-cpu-results/>

Profiling :

- [Profiler :: CUDA Toolkit Documentation](#)

JeanZay :

- http://www.idris.fr/jean-zay/gpu/jean-zay-gpu-exec_interactif.html
- `srun --pty --nodes=1 --ntasks-per-node=1 --cpus-per-task=10 --gres=gpu:1 --hint=nomultithread -A stu@gpu bash`

Deadlines

05/05 14h : revue de la TODO list
17/05 : rencontre avec mentors
25-27/05 : Hackathon

Stratégie

- OpenACC => OpenMP4/5
- fonctionnalités de vérification à grain fin dans le driver offline ?
checksums ?

DONE

- Slack => Laurent canal gpu_hackathon créé, invitations envoyées
- vérifier run sur JeanZay => Thomas, tous
- un accès pour tous au git => Romain
- un driver 'offline' sans DYNAMICO ni LMDZ => Thomas, Eliott
- un accès à Jean Zay GPU => wuu@gpu ? PSL ? sinon -A stu@gpu
- tourner avec DYNAMICO GPU : reporter 'physics_plugin' sur trunk => Ehouarn pour le moment passage sur CPU entre DYN et PHY
- tuto DYNAMICO GPU => git => Thomas
- driver offline +GPU

- mettre des bouts de GPU
- présentation 17/05 => Thomas
- ajouter des sorties (+ scripts) au driver offline
- un cas test offline qui tourne sur Jean Zay et en gitlab-CI

TODO

- introduire dans la physique un flag on_gpu
- modèle du thermique => Frédéric