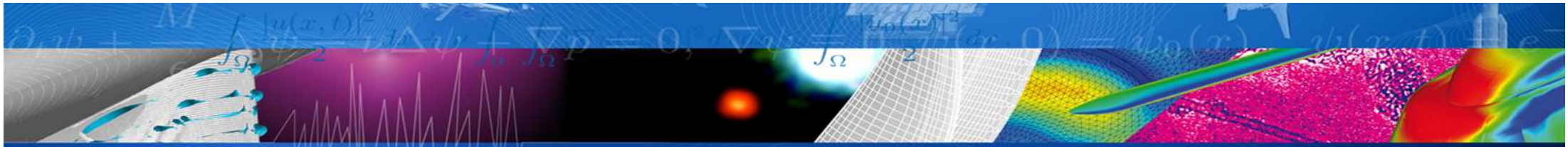


Développement de maquettes de solveurs d'écoulements compressibles en Volumes Finis non structurés pour des clusters de GPU Tesla

Matthieu Lefebvre[1], Jean-Marie Le Gouez[2], Carlos Carrascal[3]

[1] Doctorant Onera, maintenant post-doc à Princeton, département Geosciences
[2] Onera DSNA Simulation Numérique des Ecoulements et Aeroacoustique
[3] Stagiaire Onera, Master 2 Sup Galilée

Remerciements à Arnaud Renard pour l'hébergement de notre projet sur Romeo,
à Nikolay Markovskiy du service dev-tech de NVIDIA, GB

ONERA
THE FRENCH AEROSPACE LAB
return on innovation

Solveurs CFD pour maillages non structurés sur GPU

- Contexte général : l'Onera et ses partenaires
- Eléments de stratégie moyen et long terme
- Objectifs des prototypes
- Etapes des développements, modèles de données, languages de programmation, utilitaires
- Mesures de performances
- Perspectives

Contexte général : La simulation en MFN à l'Onera

Les grandes plateformes applicatives : solveurs transsoniques

•elsA : aerodynamique et aéroelasticité, mode direct et mode adjoint, optimisation automatique de forme, remaillage, très nombreux modèles de turbulence, ZDES,

•Cedre : combustion et aérothermie, écoulements diphasiques, rayonnement, thermique et couplage avec la thermomécanique structures (Zebulon)

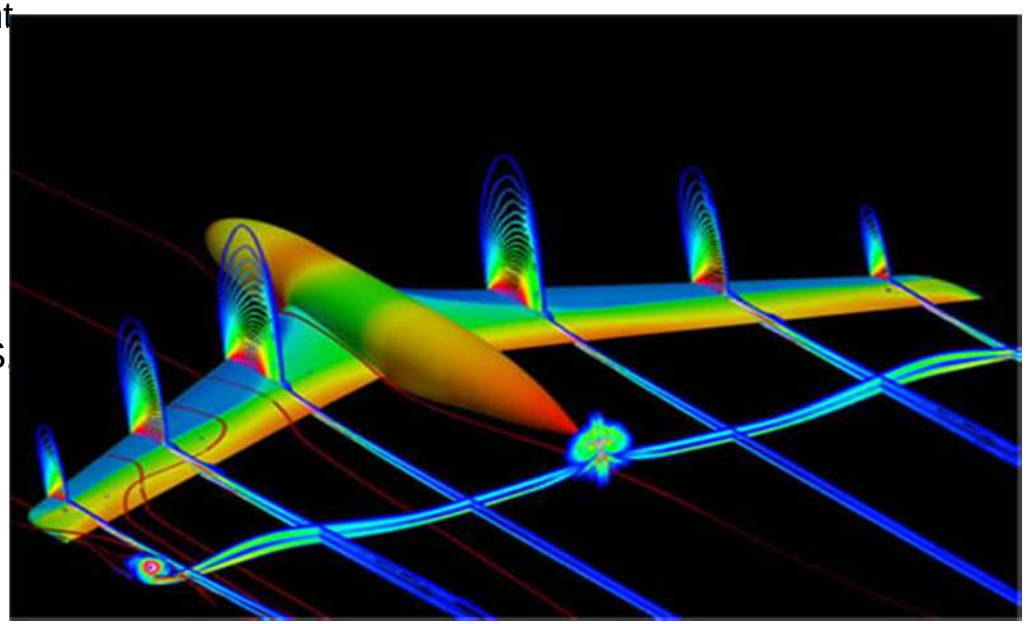
•Sabrina/Funk : aérodynamique instationnaire, optimisé pour la LES, l'aéroacoustique en perturbations, linéaires ou non

•Leurs outils de productivité des applications : géométrie, maillages recouvrants automatisés, déformations de maillage

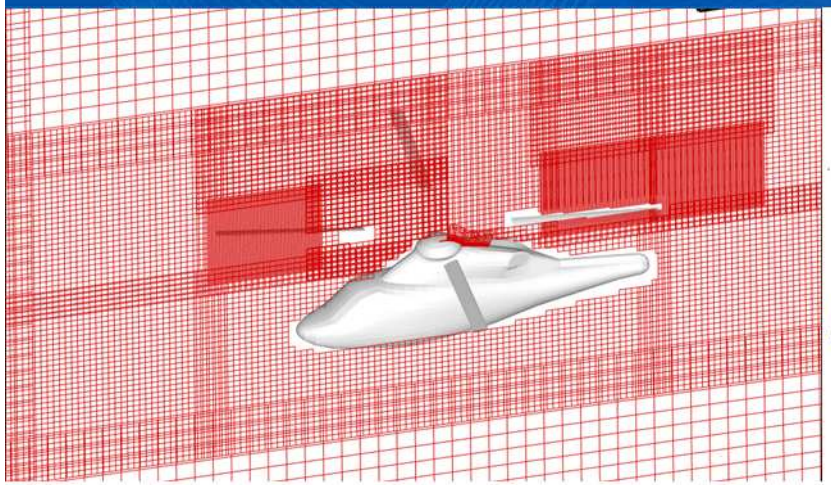
•Les couplages multi-physique : bibliothèque de couplage CWIPI, liens Open-Palm

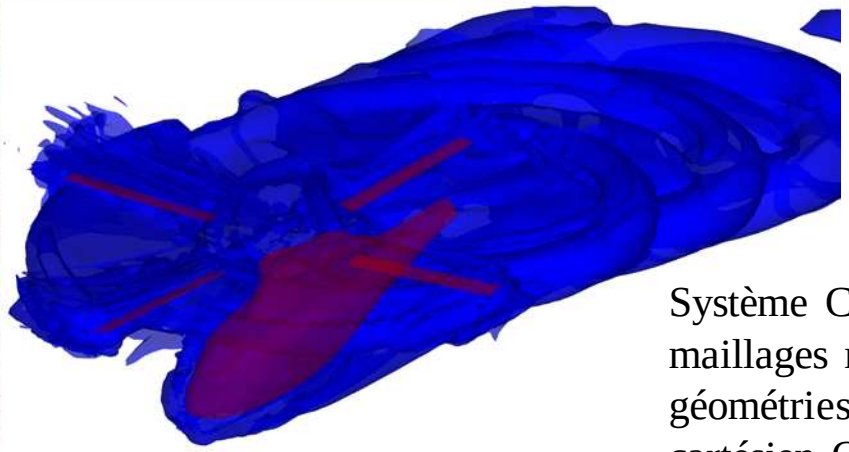
•*Centaines de milliers de lignes de code, mélanges Fortran, C++, python*

•*Usage important dans l'industrie aérospatiale*



Contexte général : La simulation en MFN à l'Onera





Système CASSIOPEE : gestion des maillages recouvrants, chimère, géométries complexes, Solveur cartésien Octree, intégration avec elsA

Les attentes des partenaires extérieurs :

•*Des domaines d'écoulement étendus :* ➔ *effets des sillages sur les composants en aval : interactions tourbillons / pales, chargements thermiques des jets sur les structures composite,*

•*Modélisation de systèmes complets et pas seulement de composants aéro : turbomachines multi-étage, couplage chambre de combustion / aérodynamique turbine, …*

•*Plus d'effets multi-échelle : modèles de paroi, représentation de détails technologiques pour améliorer l'efficacité des systèmes : rainurages de paroi, …, contrôle d'écoulements,*

•*Utilisations avancées de la CFD : modes adjoints, optimisation automatique de forme, adaptation de maillage, maîtrise des incertitudes, paramètres d'entrée définis par des pdf,…:* travaux en cours

La simulation en MFN à l'Onera

Attentes des utilisateurs internes, pour produire et valider des modèles physiques innovants :

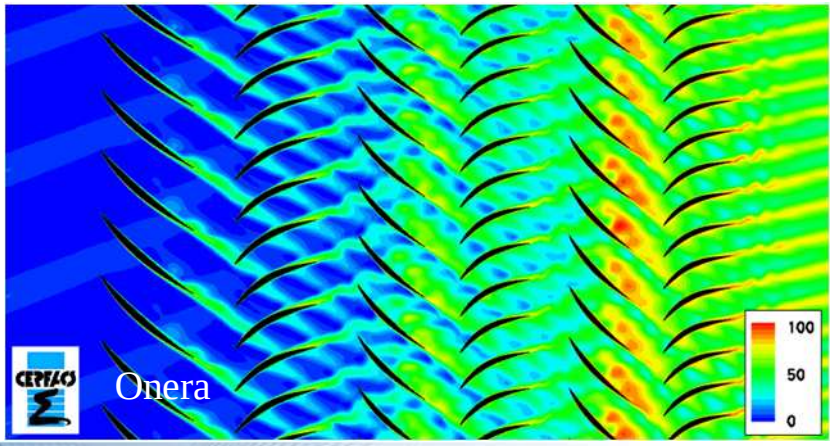
- des performances algorithmiques ,
- de nouvelles classes de méthodes numériques

➔*les méthodes numériques innovantes pourront-elles préserver leurs modèles de fermeture des interférences avec les erreurs résiduelles des schémas : dissipation, dispersion ?*

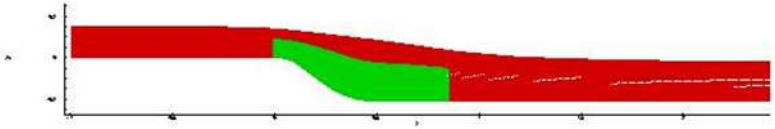
➔*plus d'efficacité et de robustesse : précision pour un nombre de ddl et une ressource CPU réduite, convergence en maillage :* ralentir la course aux milliards de cellules

➔Objectifs : longues simulations instationnaires, multi-échelle temporel, multi-physique Cedre, maillages déformables et recouvrants en mouvement arbitraire, (aéro fine LES, DES, chambres de combustion, Aéracoustique)

elsA



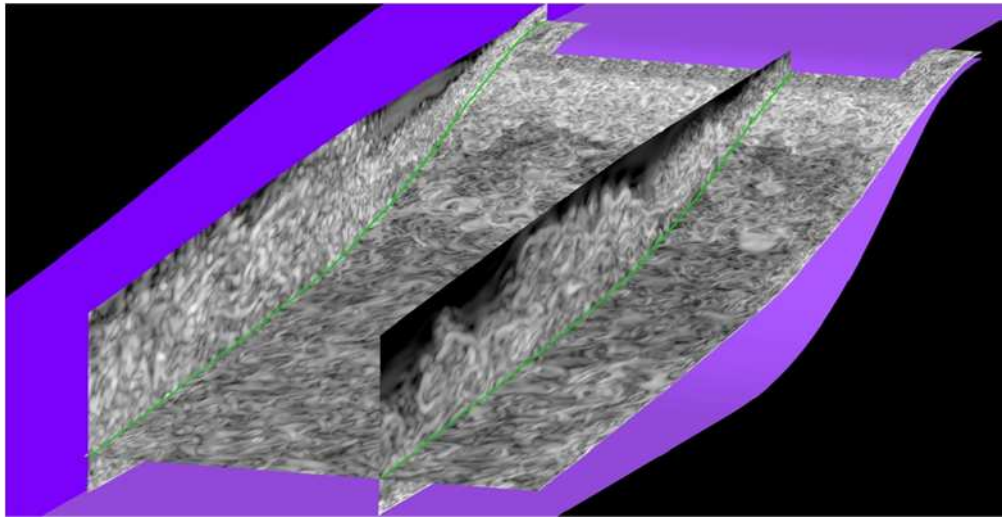
La simulation en MFN à l'Onera



Constatations :

i) Les plans de développement des plateformes de simulation "grands codes" comportent des travaux sur de nombreuses nouvelles fonctionnalités et leur interopérabilité, pas de réingénierie profonde possible ➔ risque sur la scalabilité vers le calcul pétaflopique (échéance 3-5 ans).

RANS / LES d'une manche à air



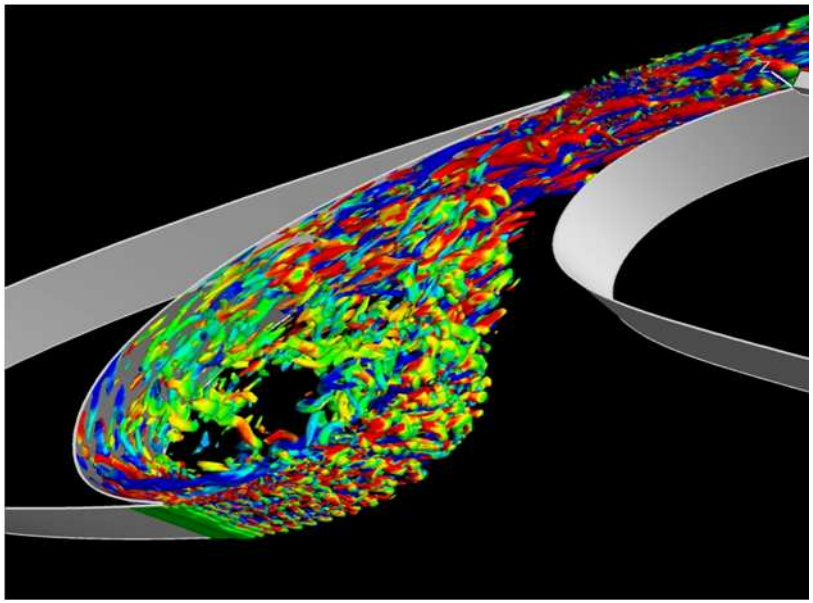
ii) La recherche innovante est active à l'Onera ; nombreux prototypes portant sur

Stratégies de maillage : - structurés
- non structurés, hybrides,
- recouvrants, auto engendrés (AMR), adaptatifs,
- à faces courbes (maillages d'ordre élevé)

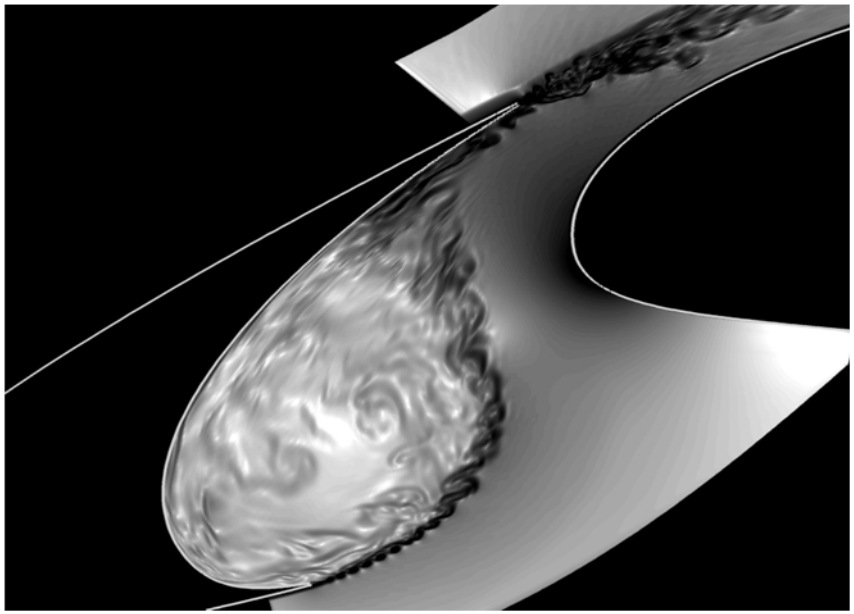
Schémas numériques et solveurs non linéaires
- DGM (perspectives vers l'adaptation h-p-M), la turbulence sous-maille VMS
- Volumes finis d'ordre élevé

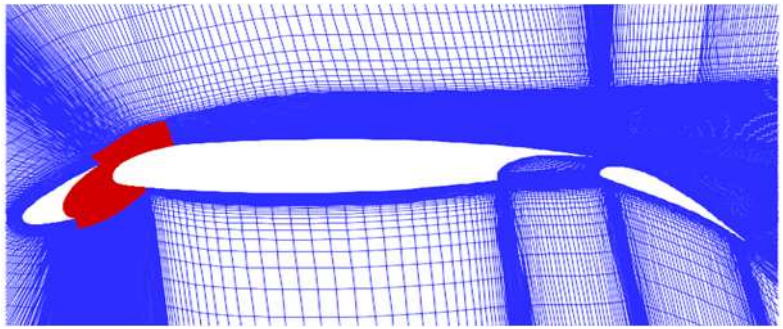
Les stratégies de couplage multi-modèle et multi-physique : niveau d'intégration des solveurs, granularité des exécutables, interopérabilité des solveurs point courant et architecture des plateformes, outils d'interpolation sur les interfaces, intégration système (en liaison avec Cerfacs)

Progression des capacités sur 5 ans : RANS / LES zonale de l'écoulement dans une cavité de bec de profil d'aile hypersustentée



2009



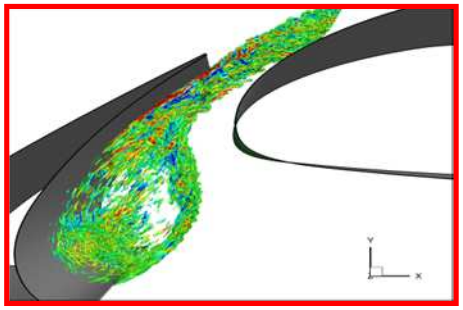


Mach 0.18
Rey 1 400 000/corde
RANS Stationnaire 2D
LES 3D 7,5 Mpts

Programme LEISA coopération DLR / Onera
Logiciel FUNK

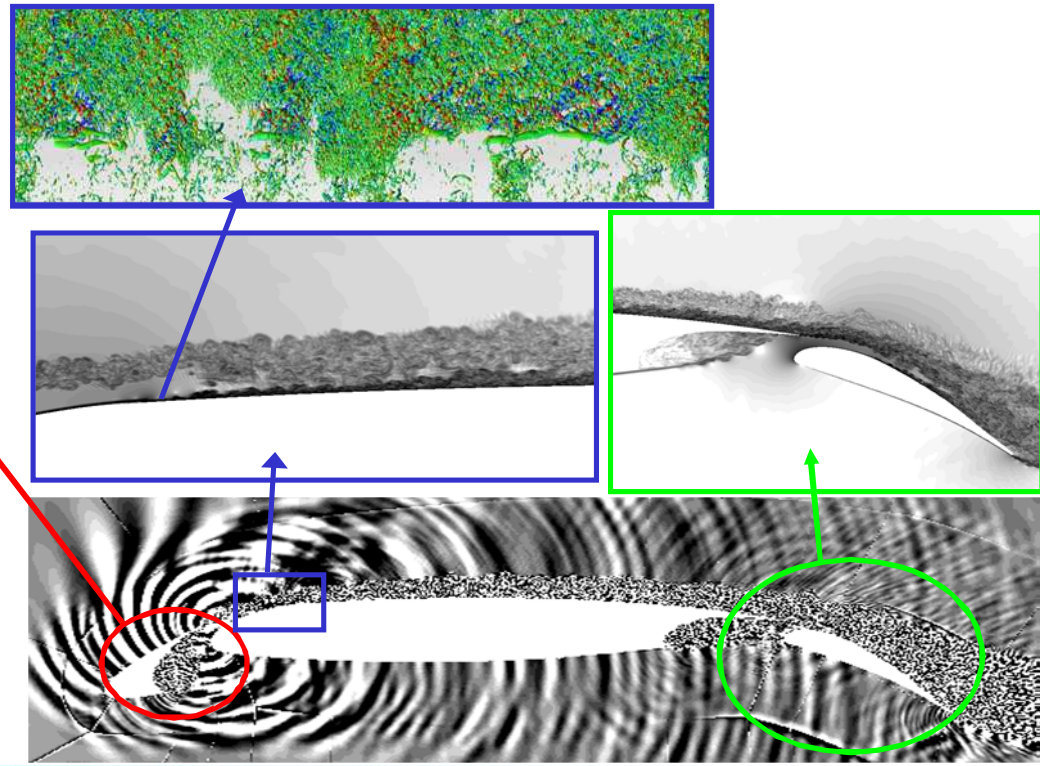
Simulation aéroacoustique d'un profil d'aile hypersustentée par Simulation des Grandes Echelles : 2014

- Profil F16 du DLR (Cas test no 6 du benchmark BANC de la NASA)
- Mesures dans F2 et AWB (coop. LEISA avec le DLR)
- Simulation de référence en complément des expériences pour validation de méthodes hybrides RANS/LES et analyse physique poussée



- **M=0.18; c = 0.3 m**
- **Re_c = 1 300 000.**
- LES résolue à la paroi:
  Δx^+~30-40, Δy^+~1, Δz^+~10

- **Solveur Volumes Finis *FUNk***
- ***Fortement optimisé sur architecture CPU : MPI / OpenMP / vectoriel***
- **Ressources CPU (~70ms de signal):**
  - **Calculateur JADE (CINES)**
  - **N_{xyz}~ 2 600 000 000 pts**
  - **4096 coeurs / 10688 domaines**
  - **T_{CPU}~ 6 200 000 h**



Projets DARI *c20142a7049 & 2013-2a7049*)

Objectifs : prise en compte accrue du HPC

Dans le développement de solveurs prototypes, marginalement dans les grands codes (plutôt dans leur environnement)

FUNK : optimisation poussée MPI / OpenMP, cache blocking, vectorisation sur des directions topologiques de maillage particulières : code structuré i,j,k, référence en performances pour la LES

AGHORA : méthode DG, géométrie d'ordre élevé, fermeture sous-maille spectrale, parallélisation mixte MPI / OpenMP, collaboration démarre avec l'INRIA : STAR-PU
+ solveur isogéometrique (base éléments par Nurbs)

NEXTFLOW : volumes finis d'ordre élevé, MPI / OpenMP / Cuda

Proposer des architectures logicielles et méthodologies sur la base d'une analyse des hierarchies d'accès aux mémoires, aux coeurs de calculs, aux fermes de coprocesseurs,

Specifier des solveurs modulaires sur des maillages partitionnés, des modèles de données adaptatifs et hiérarchiques, des méthodes pour réutiliser (réécrire, simplifier) les modules des grands codes,

Etendre la base d'outils de couplage et d'utilitaires de pilotage d'applications complexes,

Participer aux initiatives pour des langages de haut niveau en Mécanique des Fluides ((DSL)

Nouvelles méthodes de maillages possibles : Cassiopée hiérarchie dans les maillages et les liens CAO

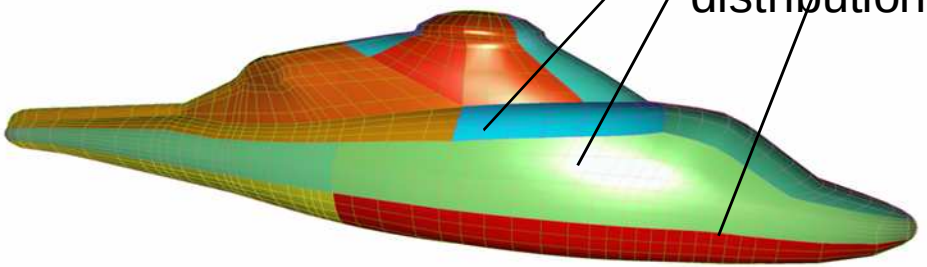
- Macro maillage/raffinement embarqué

Raffiné sur chaque processeur

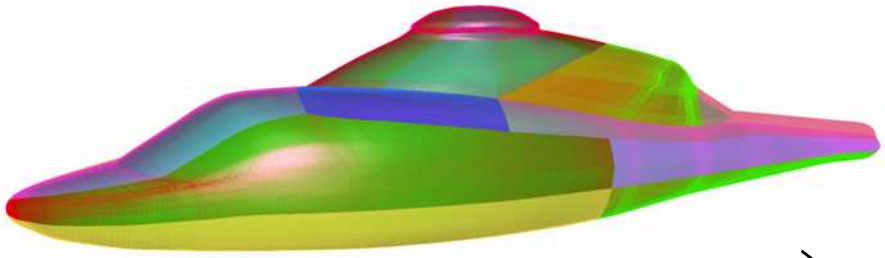
P1 P2 P3

Maillage engendré

distribution



Maillage réellement calculé



Extraction embarquée

Le maillage dense n'est jamais manipulé ou stocké

AGHORA – High Accuracy Navier-Stokes Software Prototype for HPC

Scientific challenge : *High accuracy and efficiency for complex turbulent flow simulations with Navier-Stokes on HPC plate-formes*

Aghora software prototype: CFD 2020

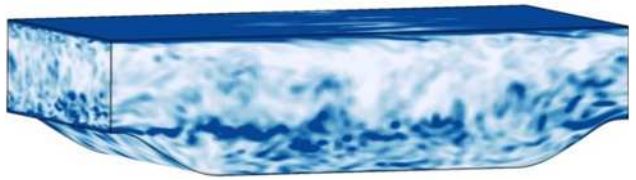
- *TRL 3 in 2015 for internal and external aerodynamics (Safran, Airbus)*
- Adaptative models : high-order schemes and meshes, multi-level models
- Efficient HPC programming on heterogeneous architectures (with Inria)
- Management of operational uncertainties (stochastic methods)

Physical modeling and numerics

-DG modal or nodal approach on unstructured polyhedra
-RDS development under way (with Univ. of Zürich)
-Internal and external flow configurations
-Different levels of turbulence modeling : RANS, DES, LES, DNS

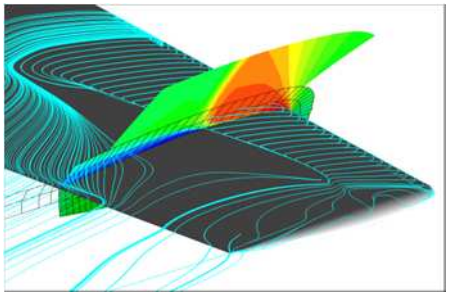
Associated projects : PRF Aghora (Onera projects), European projects IDIHOM, ANADE, Possible new H2020 European project TILDA, French project DGCIS /ELCI, FP7 UMRIDA,

High-Order CFD workshop on various Navier-Stokes test cases

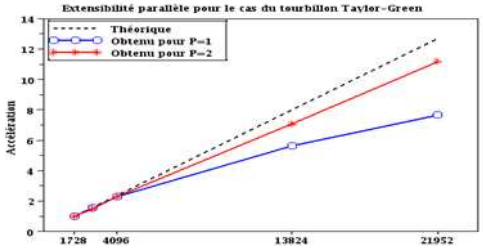


Aghora – Periodic hill
DNS computation (DG O4)

V. Couaillier, F. Renac, M.de la Llave Plata,,J.B. Chapelier, E. Martin, M.C. Le Pape "Turbulent Flow Simulations with the High-Order DG Solver Aghora", AIAA -2015-0058



Aghora – Onera-M6 wing
Transonic RANS kω computation

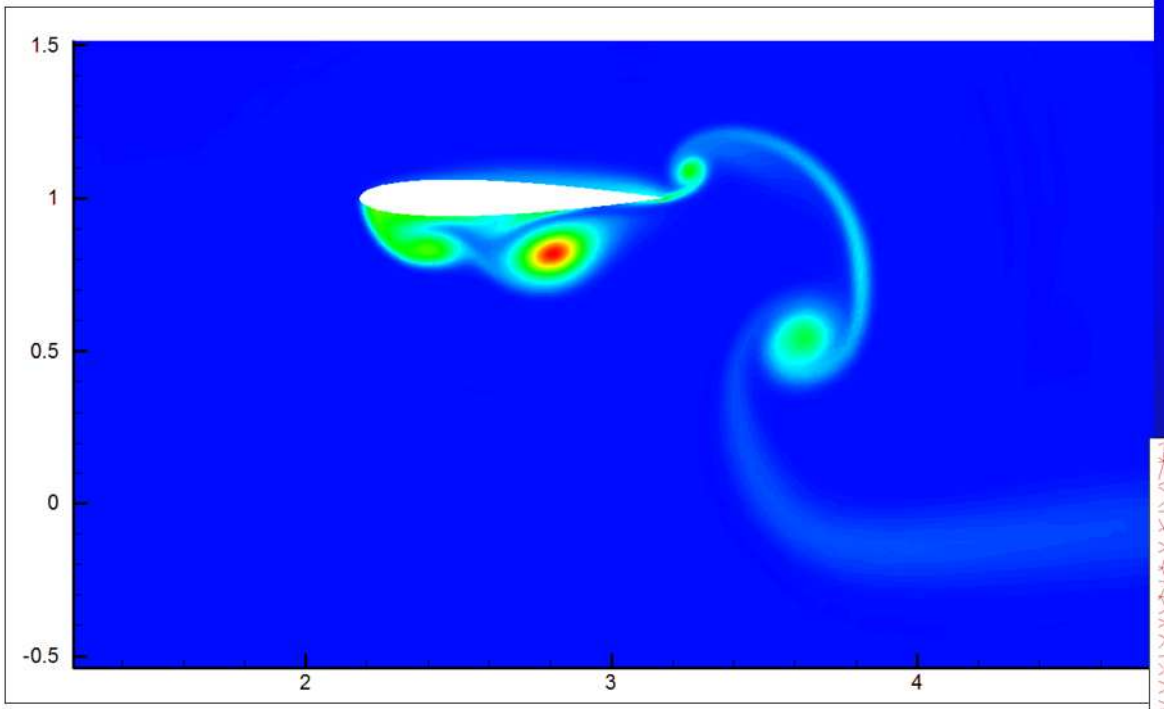


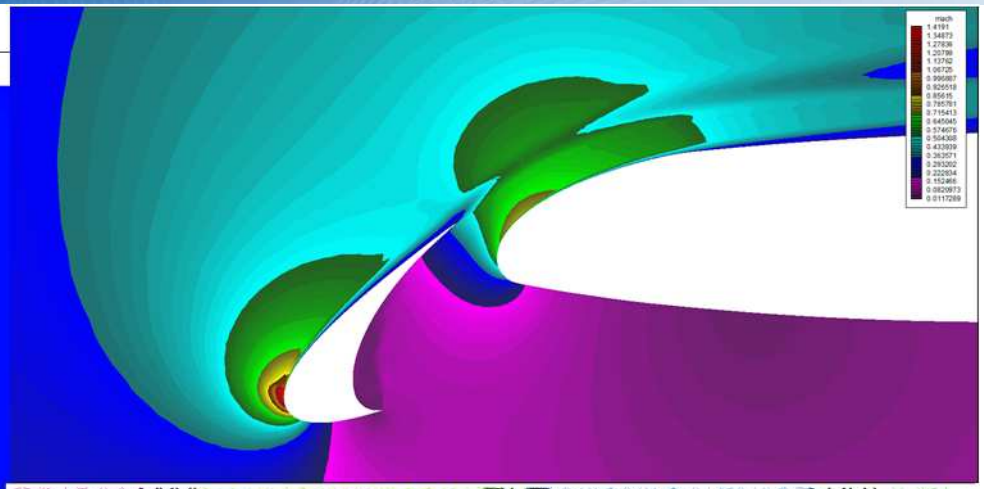
Aghora – Better strong scalability with high polynomial degree

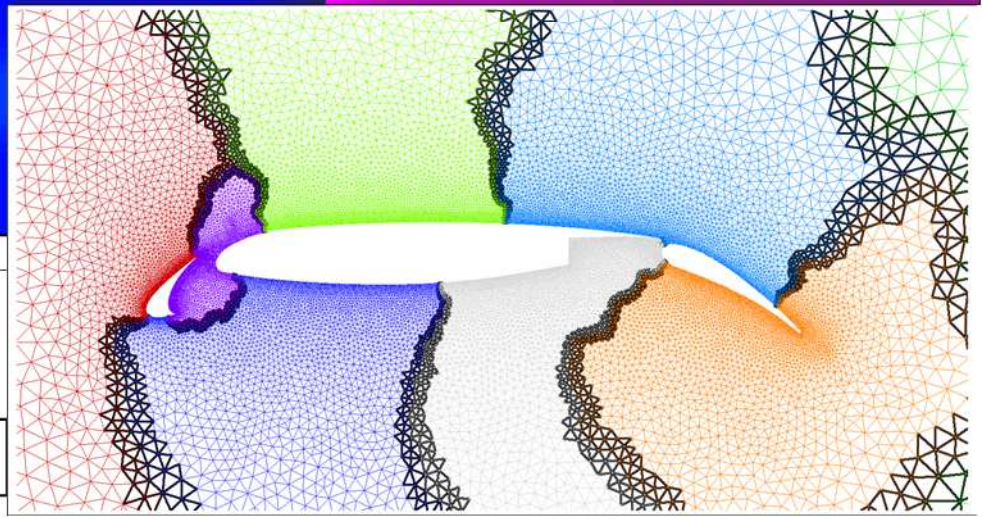
ONERA
THE FRENCH AEROSPACE LAB

NextFlow : **High Order CFD Workshop** **Case 2.3 Heaving and Pitching profile**

RANSE High Lift flow

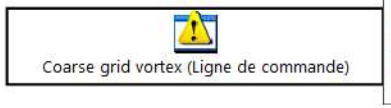


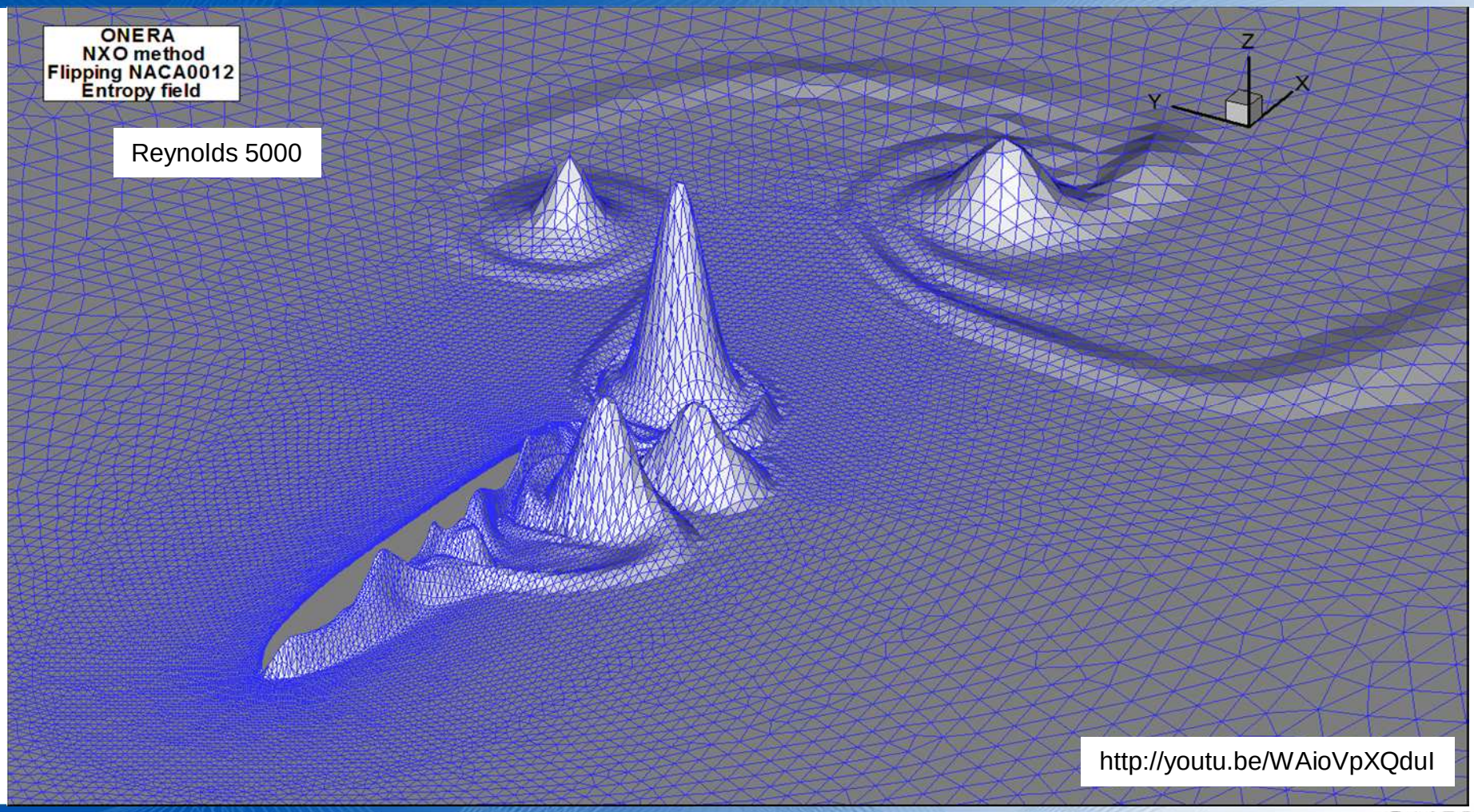




http://youtu.be/41Fnis_sc-w

Entropy field Reynolds 5000




ONERA
NXO method
Flipping NACA0012
Entropy field
Reynolds 5000
Z
X
Y
http://youtu.be/WAioVpXQdul

NextFlow : prototype générique en volumes finis d'ordre élevé sur GPU

contribution à la maîtrise des architectures hardware hétérogènes

Utilise une méthode spatiale d'ordre élevé en Volumes Finis non structurés : NXO

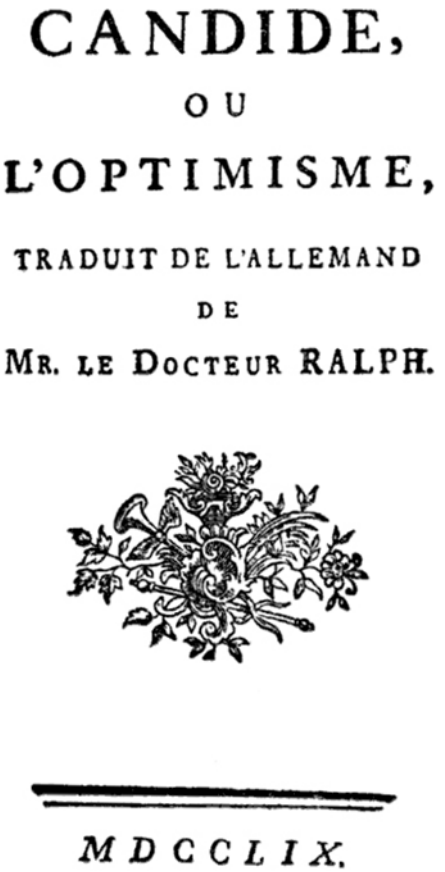
A partir d'une base Fortran / MPI, qui utilise des structures de données complexes sur des grilles partitionnées, portage mixte OpenMP Fortran et C / Cuda à travers une interface qui transpose les tableaux de travail, gère les pointeurs sur les mémoires GPU.

1ère version : structure de données i,j,k : choix du language, des outils,

2ème version : partitionnement en blocs du maillage fin global, allocation aux blocs de threads,

3ème version : modèle hiérarchique, maillage grossier d'ordre élevé (frontières courbes) raffiné dynamiquement sur le GPU : assure un accès coalescent aux données en mémoire globale dans toutes les phases des algorithmes (noyaux de calcul par éléments, par faces, par noeuds)

4ème version (en cours) : modèle 2,5D périodique en envergure, vectorisé sur CPU et parallélisme de données selon la 3ème dimension homogène


CANDIDE,
OU
L'OPTIMISME,
TRADUIT DE L'ALLEMAND
DE
Mr. LE Docteur RALPH.
MDCCLIX.

« – Cela est bien dit, répondit Candide, mais il faut écrire notre Cuda - »

Cuda : permet aux développeurs non « programmeurs–Ninja » un apprentissage de la programmation avancée, éco-responsable, par exemple :

Gestion des ressources critiques des emplacements mémoire des multi-processeurs

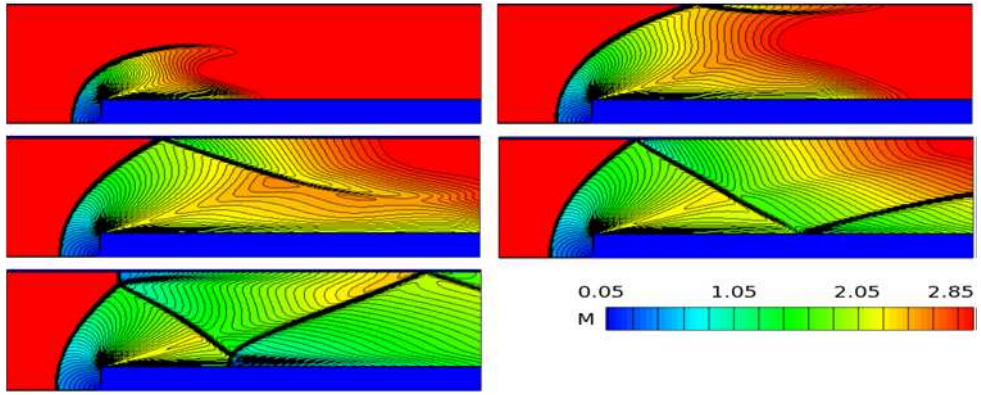
leur occupation est bloquée par le nombre de registres utilisés dans un noyau : il y a bien un coût associé à cette mémoire près des unités de calcul, leur nombre est limité, l'algo doit s'adapter. En retour : gain spectaculaire de performances

1ère version structurée i,j,k : utilisée pour choisir les languages, API, outils testés :

CUDA C/Fortran, OpenCL

PGI Accelerator

Solveurs 2D and 3D

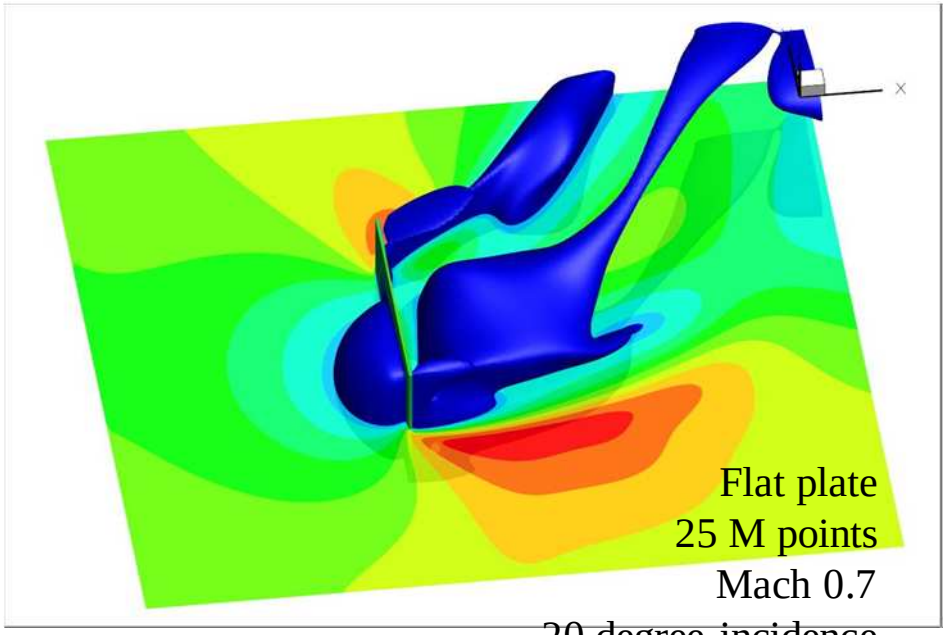


1ère version : Cartesian Euler : Multi-GPUs Communication Scheme

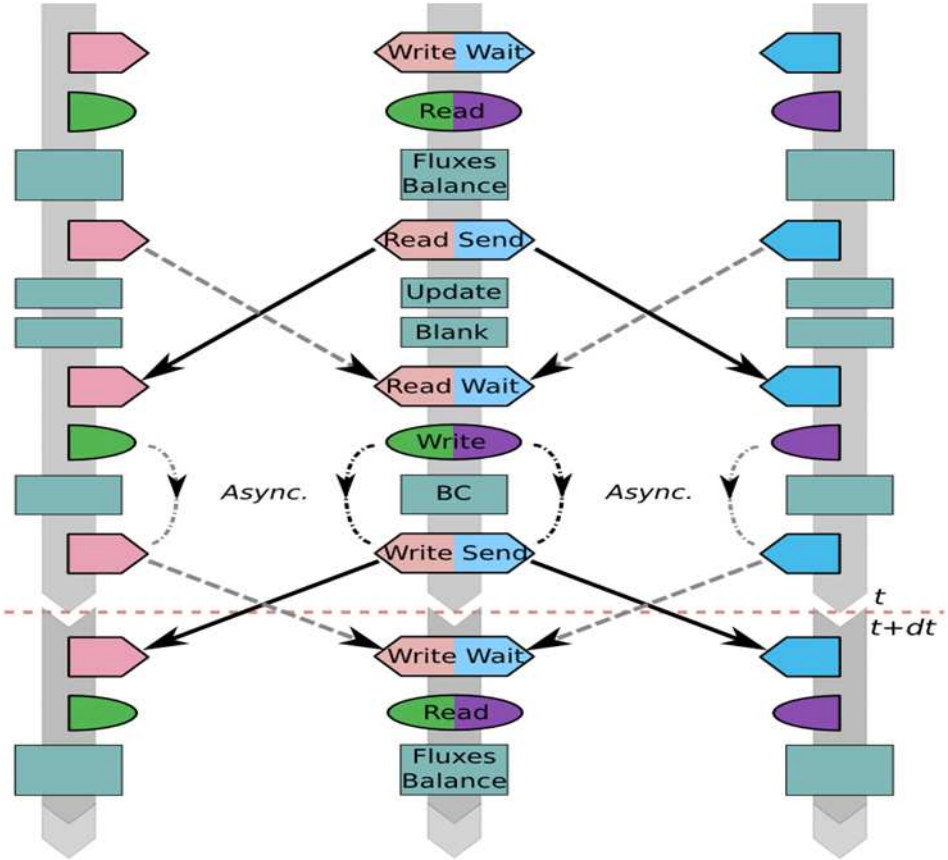
CUDA C model preferred

2 Fermi GPUs

collaboration through the pthread library

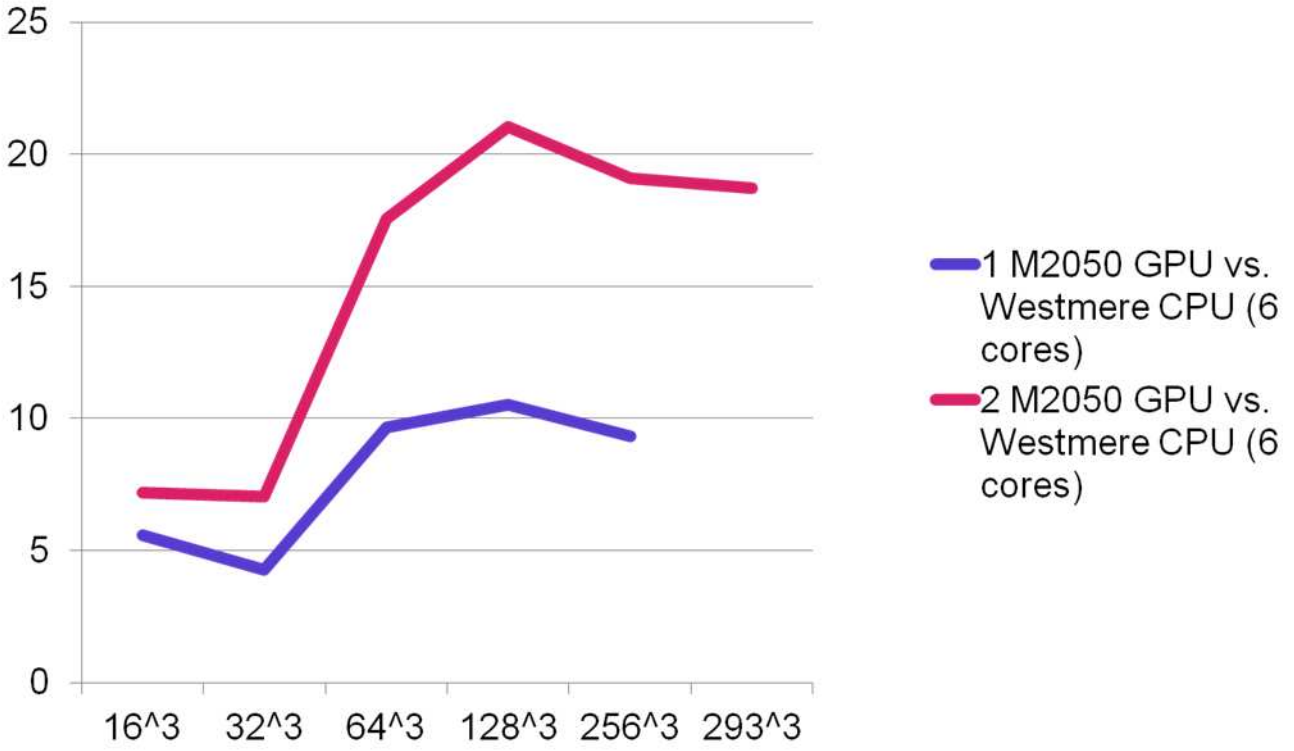


Flat plate
25 M points
Mach 0.7
20 degree incidence



3D Experiments – Speedup and Scalability Results

Speedup



Problem size

Speedup 1 M2050 vs. 2 M2050 : up to 2.11

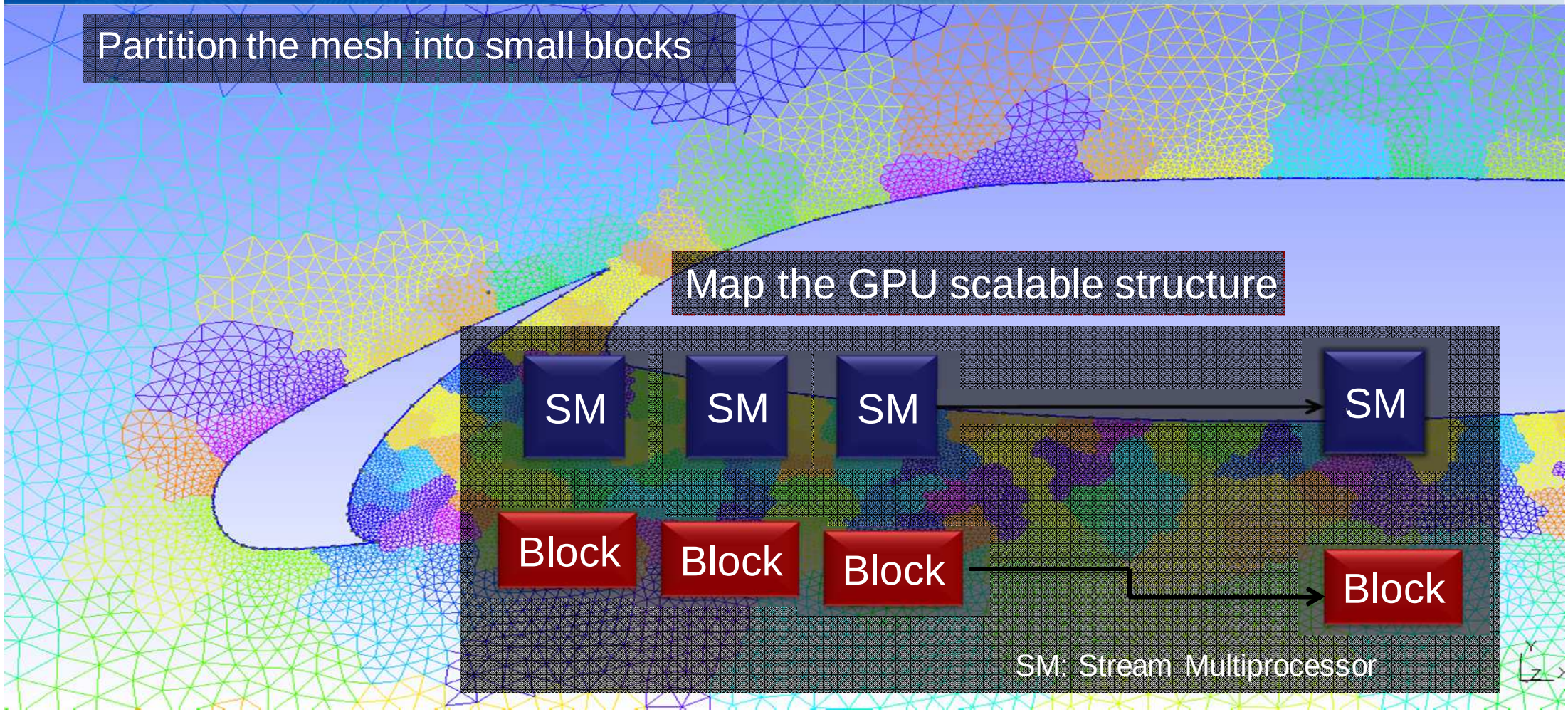
Challenge: Accelerating Non “GPU Compliant” Codes

NXO

- Unstructured grids
- Large stencils (up to the third neighbor to compute the fluxes for each cell interface, very high memory stress)
- Reaches 4th-order spatial accuracy
- However algorithmic efficiency is hampered by the numerous indirect memory accesses through the arbitrary connectivity lists accessed at successive stages of the algorithm (cell-,face-, node-, stencil-based descriptions).

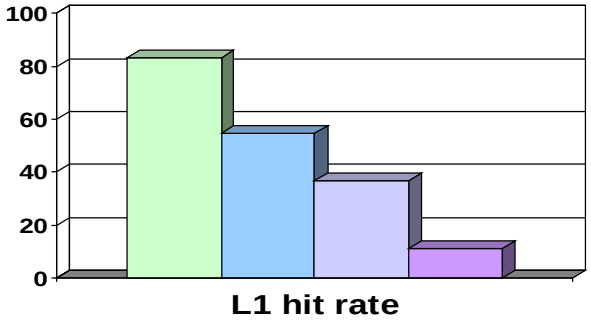
On a GPU such non consecutive memory fetches are penalizing

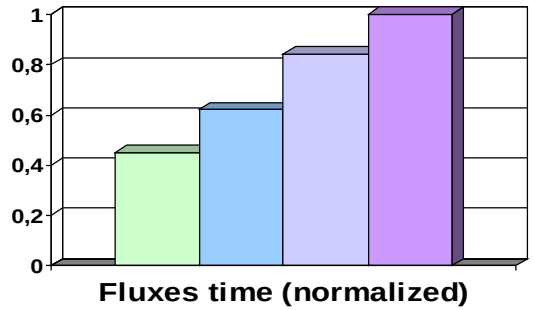
1st Approach: Block Structuration of a Regular Linear Grid

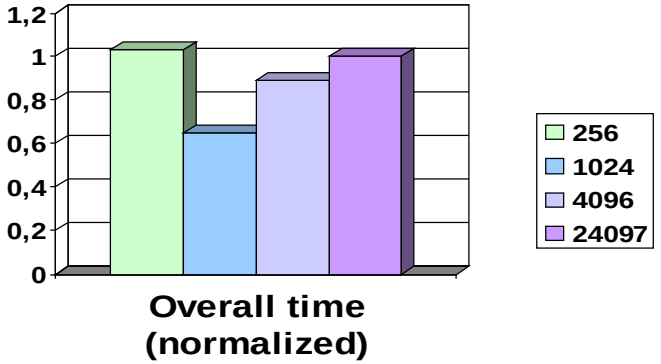


Advantage of the Block Structuration

- Bigger blocks provide
  - Better occupancy
  - Less latency due to kernel launch
  - Less transfers between blocks
- Smaller blocks provide
  - Much more data caching

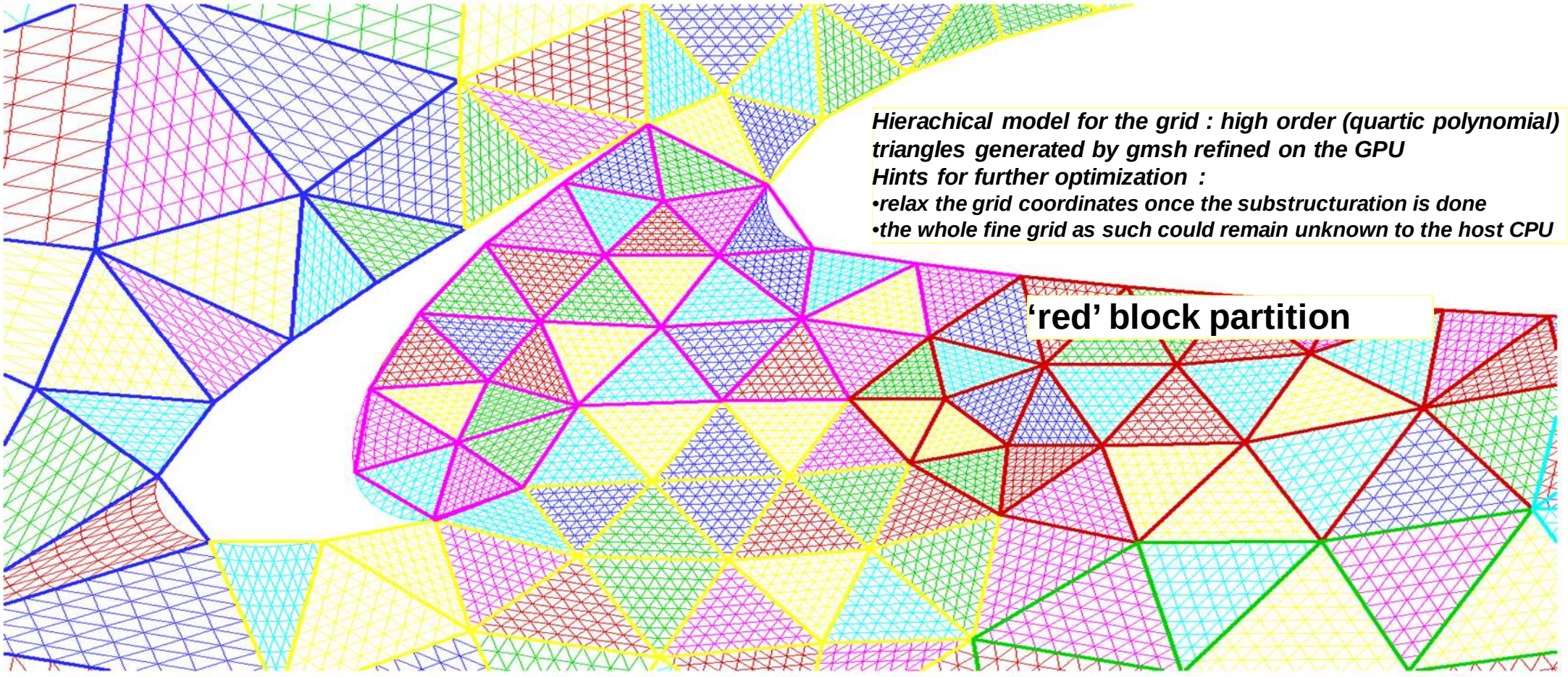






- **Final speedup wrt. to 2 hyperthreaded Westmere CPU: ~2**

NXO-GPU Phase 2 : Imposing an Inner Structuration to the Grid
(inspired by the 'tesselation' mechanism in surface rendering)

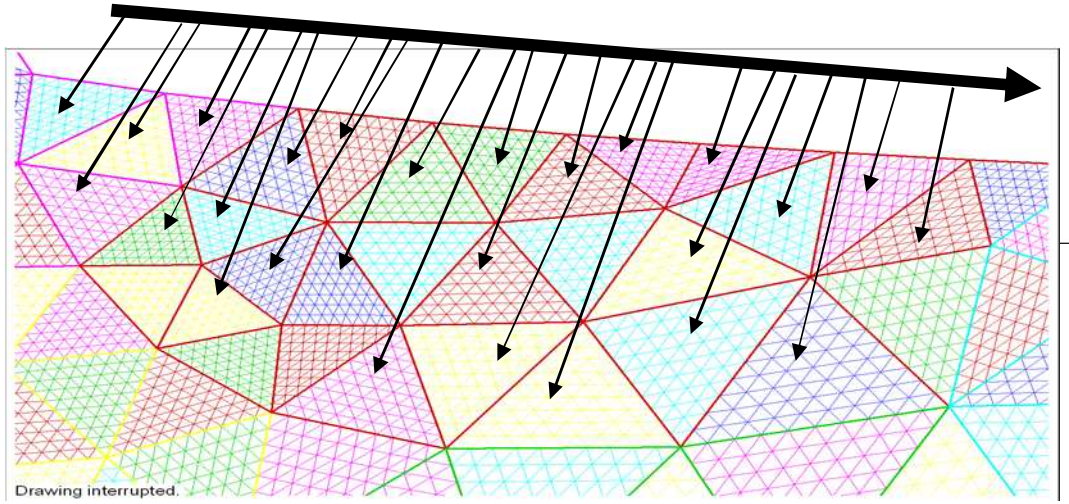


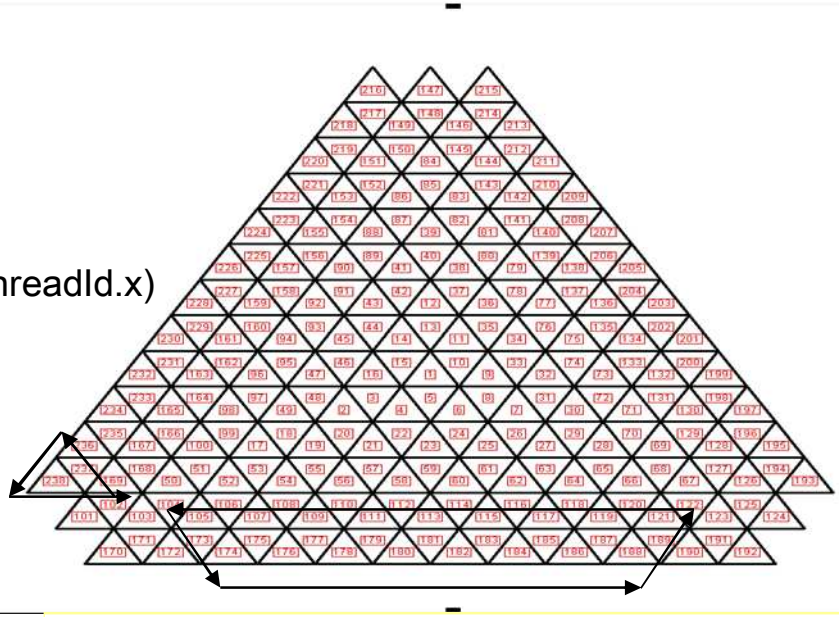
NXO-GPU Phase 3 : Imposing an Inner Structuration to the Grid

Unique grid connectivity for the algorithm
Unique subsets of cell groups for communications

Optimal to organize data for *coalescent memory access*
during the algorithm and communication phases

Each coarse element in a block is allocated to an inner thread (threadId.x)



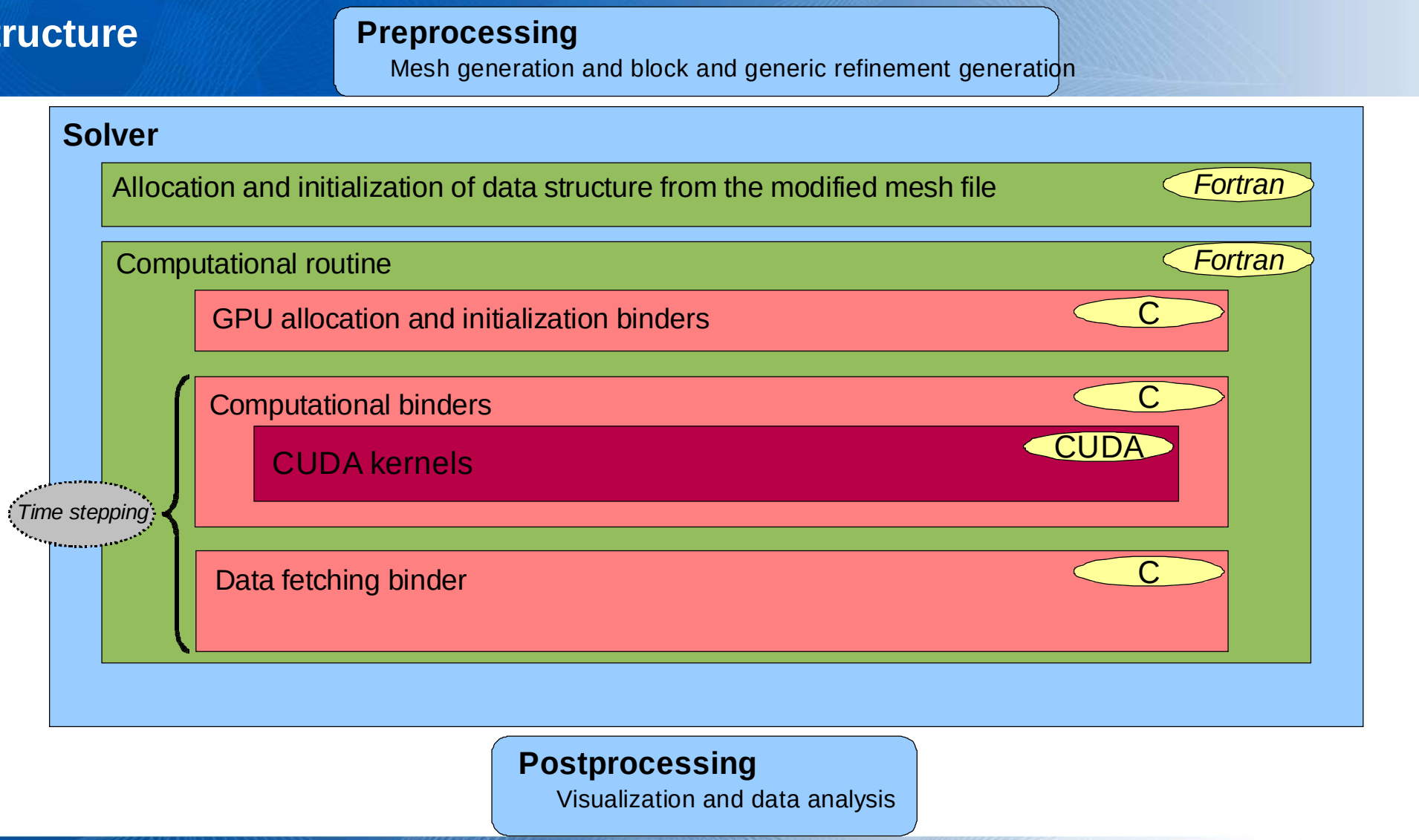


Sub-structured interface vector for communication

To improve the percentage of coalescent communication :

- Renumber the coarse cells in a partition
- Change the orientation (permutaion of the sides)

Code structure

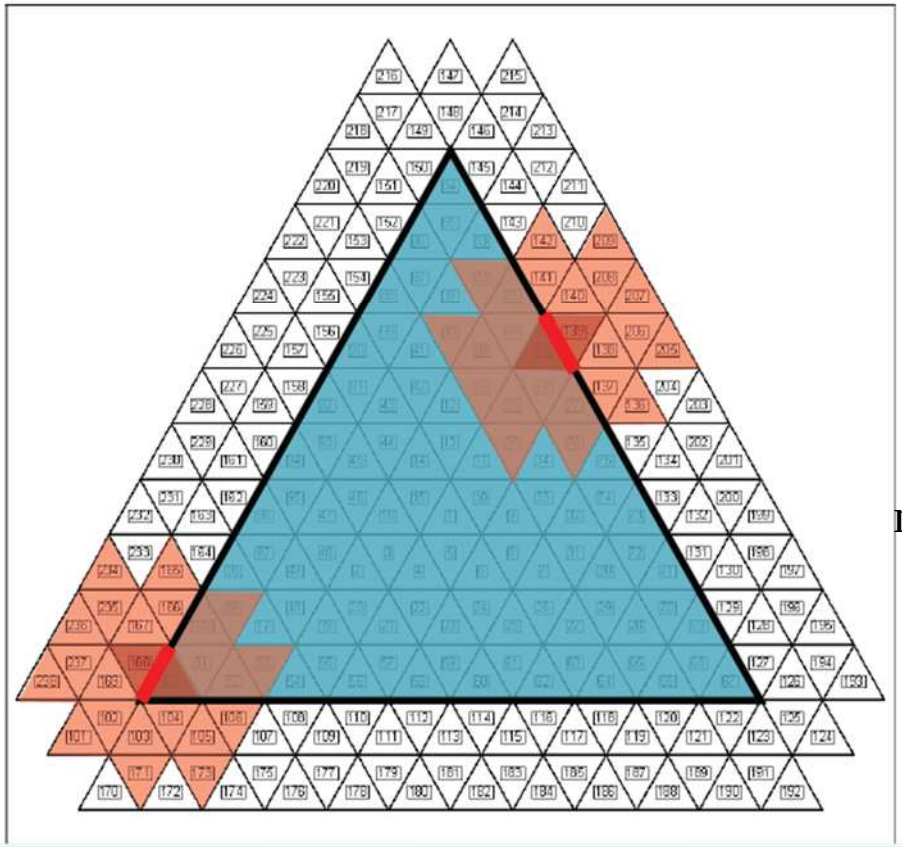


Coupling mechanisms :

Identify ghost cells with real fine cells from neighbor coarse cells (transfer its metrics)

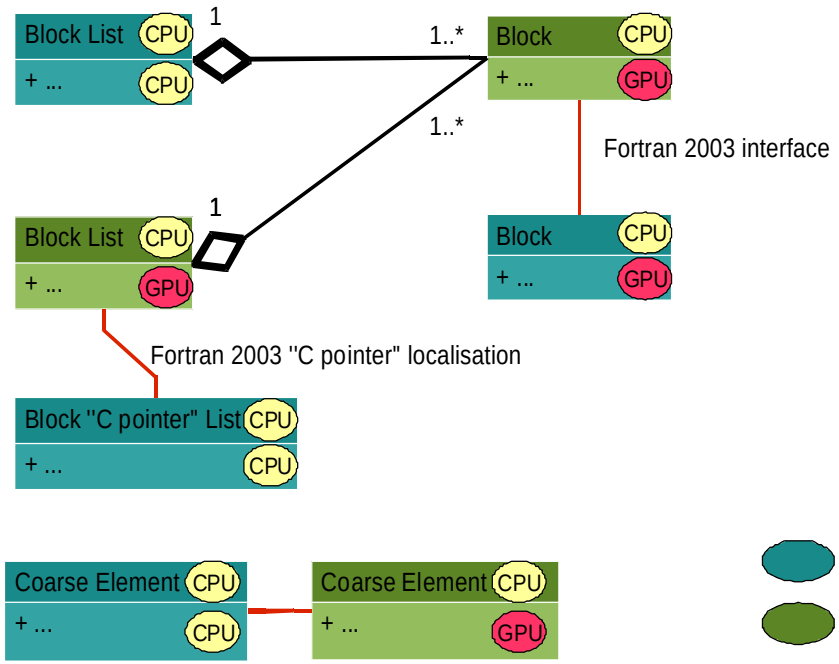
or :

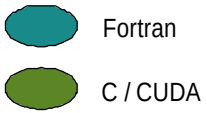
do an overset grid interpolation to obtain the volume average of the conserved variables on the extruded ghost cells

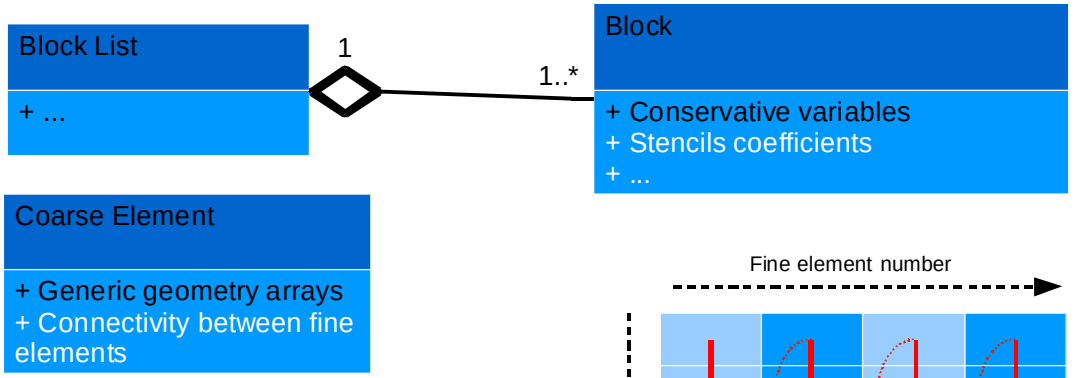


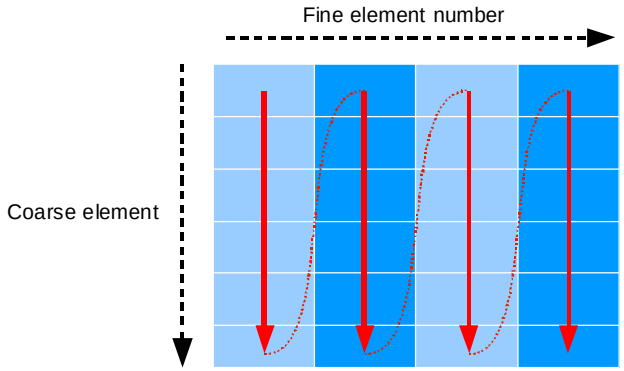
Reference element to be mapped on curvilinear cells

Code structure









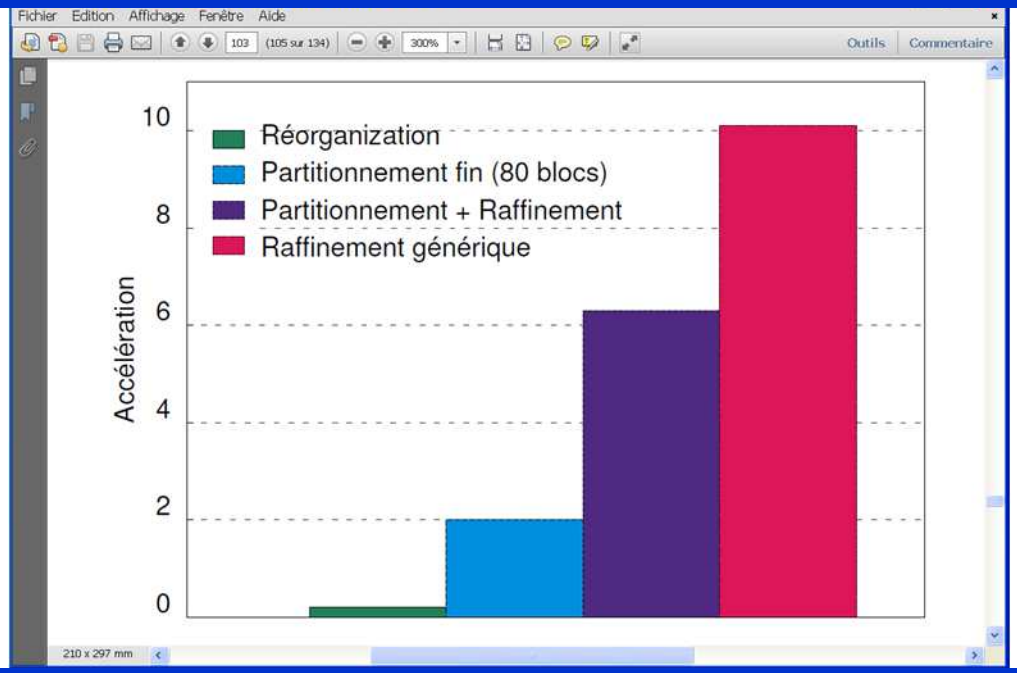
Array filling : face

Code structure — Language interoperability

```c
typedef struct {
  int number_elements;
  int number_faces;
  int size_stencil;

  double *wall_distance;
  double *dissipation_rate;
  // ...
  double *flu_u;
  double *flu_v;
  // ...
  double *producted_kinetic_energy;
  double *pseudo_timestep;
  double *ro;
  double *sfx;
  double *sfy;
  double *temperature;
  double *viscosity;
  double *vol;
  //...
  int *face_vertices;
  int *index_element_faces;
  int *index_face_elements;
  int *number_elements_stencils;
  int *stencil_indexes;
  int *index_stencil_faces;
  // inter coarse element communications
  int *message_src;
  // Boundaries conditions
  int number_adherent_faces;
  int number_open_faces;
  int *adherent_faces;
  int *open_faces;
} block_data_t;
```

```fortran
TYPE block_data_t
  integer(C_INT) :: number_elements
  integer(C_INT) :: number_faces
  integer(C_INT) :: size_stencil

  real(C_DOUBLE), dimension(:), allocatable :: wall_distance
  real(C_DOUBLE), dimension(:), allocatable :: dissipation_rate
  ! ...
  real(C_DOUBLE), dimension(:), allocatable :: flu_u
  real(C_DOUBLE), dimension(:), allocatable :: flu_v
  ! ...
  real(C_DOUBLE), dimension(:), allocatable ::
producted_kinetic_energy
  real(C_DOUBLE), dimension(:), allocatable :: pseudo_timestep
  real(C_DOUBLE), dimension(:), allocatable :: ro
  real(C_DOUBLE), dimension(:), allocatable :: sfx
  real(C_DOUBLE), dimension(:), allocatable :: sfy
  real(C_DOUBLE), dimension(:), allocatable :: temperature
  real(C_DOUBLE), dimension(:), allocatable :: total_energy
  real(C_DOUBLE), dimension(:), allocatable :: viscosity
  real(C_DOUBLE), dimension(:), allocatable :: vol
  ! ...
  integer(C_INT), dimension(:), allocatable :: face_vertices
  integer(C_INT), dimension(:), allocatable :: index_element_faces
  integer(C_INT), dimension(:), allocatable :: index_face_elements

  integer(C_INT), dimension(:), allocatable ::
number_elements_stencils
  integer(C_INT), dimension(:), allocatable :: stencil_indexes
  integer(C_INT), dimension(:), allocatable :: index_stencil_faces
  // inter coarse element communications
  integer(C_INT), dimension(:), allocatable :: message_src
  ! Boundaries conditions
  integer(C_INT) :: number_adherent_faces
  integer(C_INT) :: number_open_faces
  integer(C_INT), dimension(:), allocatable :: adherent_faces
  integer(C_INT), dimension(:), allocatable :: open_faces
END TYPE block_data_t
```

Version 3 : Measured efficiency on Tesla 2050 and K20C (with respect to 2 Cpu Xeon 5650, OMP loop-based)



Flop count : around 80 Gflops/K20C

These are valuable flop, not Ax=b flop but Riemann solver flop with high order (4th, 5th) extrapolated values, characteristic splitting, … : requires a very high memory traffic to permit theses flops

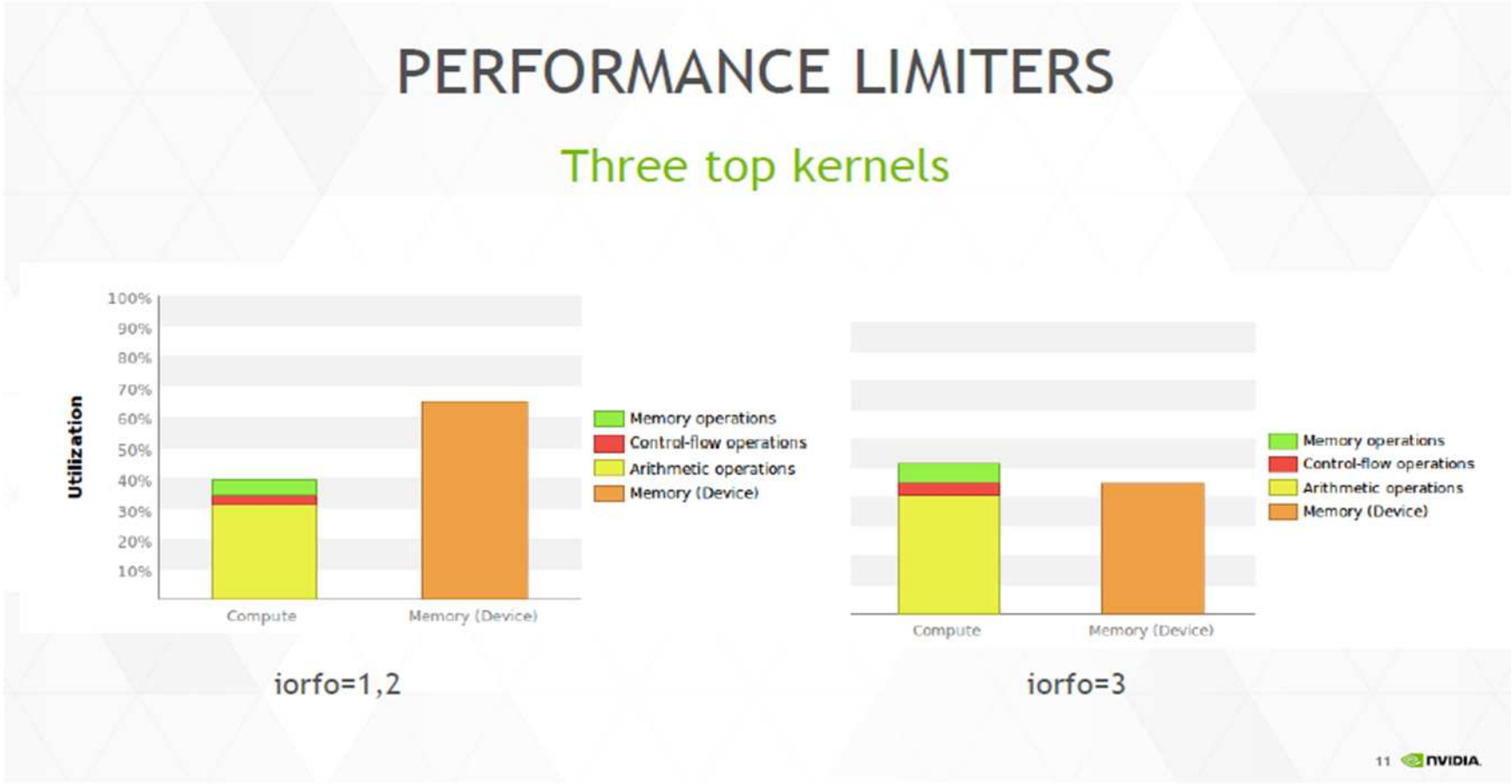
Thanks to the NVIDIA dev-tech department for their support, " my flop is rich"

Results on a K20C : Max. Acceleration = 38 wrt to 2 Westmere sockets, very good scaling from the C2050 with the number of cores (2000 / 480), due to a lower pressure on registers on Kepler? This is also due to a poor CPU optimization

Improvement of the Westmere CPU efficiency : OpenMP task-based ➔ the blocks are refined on the CPU also, then the K20C GPU / CPU acceleration drops to 13 (1 K20c = 150 Westmere cores)

In fact this method is memory bounded, and GPU bandwidth is critical. More CPU optimisation needed (cache blocking, vectorisation ?)

Version 4 : 2.5D periodic spanwise (cshift vectors), MULTI-GPU / MPI



Version 4 : Full CPU vectorisation (all variables are vectors of length 256 to 512) in the 3rd homogeneous direction, full data parallel Cuda kernels coalesced

RESULTS

Comparison with Original code (1 partition)

Original	Current optimizations
Compute : 85.0936872959137 12.8738024234772	Compute : 84.5988802909851 5.95152688026428
Newtime : 2.962398529052734E-002 3.664016723632812E-003	Newtime : 2.988314628601074E-002 3.657817840576172E-003
Newiter : 0.201013803482056 2.434015274047852E-002	Newiter : 0.196562051773071 3.784275054931641E-002
Dtloc : 7.01542568206787 0.916594982147217	Dtloc : 7.00739192962646 0.480051517486572
Timeres : 1.90137290954590 0.292651176452637	Timeres : 1.89031934738159 0.300009727478027
Fluxes : 69.2380857467651 9.93227958679199	Fluxes : 68.8473780155182 4.28139758110046
Fluxbal : 3.26401877403259 0.913108825683594	Fluxbal : 3.21450090408325 0.439968824386597
Update : 3.44401168823242 0.791139841079712	Update : 3.41265749931335 0.408583641052246
Messages : 1.347064971923828E-004 2.384185791015625E-005	Messages : 1.873970031738281E-004 1.502037048339844E-005

8 IVB cores vs K40(ECC ON, GPU Boost ON)

12 NVIDIA

After some GPU optimizations : Acceleration 14 with respect to 8 IVB cores, GPU memory Bandwith 150 GB/s

For LES, GPU memory size not too much of a problem : 8 million cells stored on a K40