

Visualisation des Maillages AMR Tree-Based

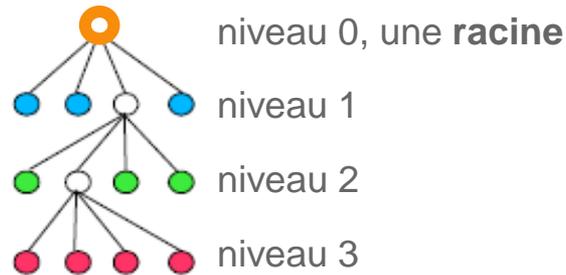
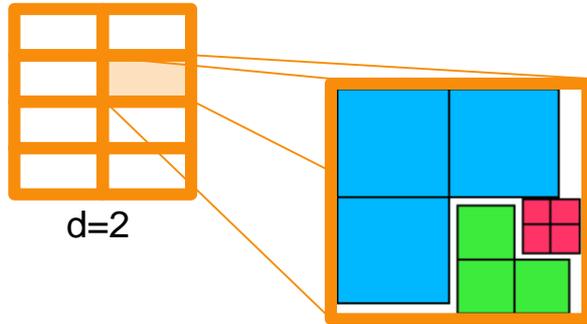
Guénolé HAREL, Jacques-Bernard LEKIEN, Philippe Pébaÿ

CEA, DAM, DIF, F-91297 Arpajon, France

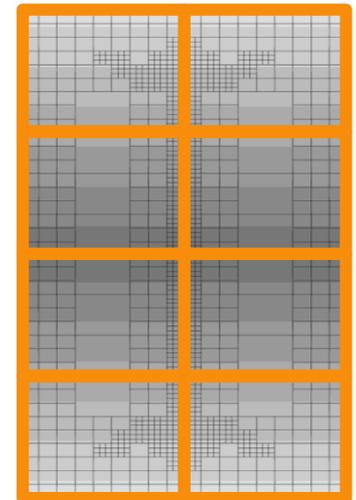


DEFINITION : AMR TREE-BASED

A partir d'une grille rectilinéaire (non uniforme), on définit les **mailles racines**.



- d , nombre de dimension spatiale : 2 (quad) ou 3 (hexa)
- f , facteur de raffinement : 2 ou 3
- ⇒ 1 maille grossière (**mère**) = f^d mailles plus fines (**filles**)
- Maille non raffinée est une **feuille**.



$d=2, f=3$

- Parallélisme : répartition (non contigüe) des racines
- Masque, interface...

Le CEA a développé :

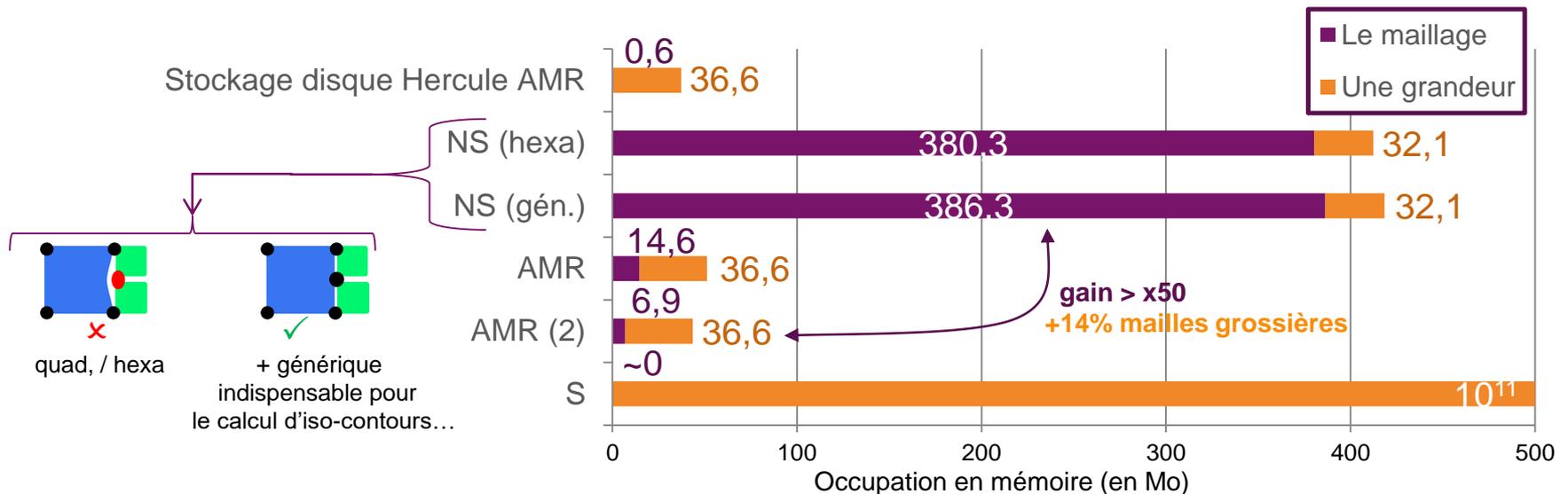
- en 2008 : un **stockage ultra-compact** dans *Hercule* (bib. E/S de nos codes).
- depuis 2012: une **représentation compacte** pour l'analyse et la visualisation

interactive : *vtkHyperTreeGrid* dans  *Visualization Toolkit*



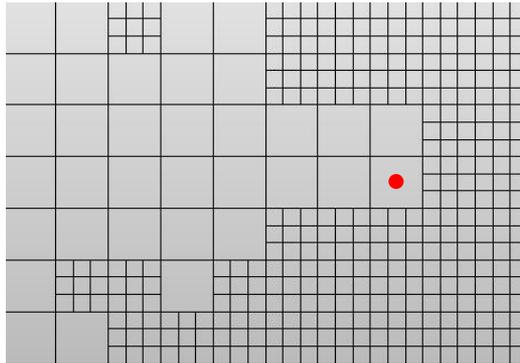
initialisation itérative \Rightarrow l'occupation mémoire dépend de l'ordre de parcours à l'initialisation

➤ Cas 4 millions de mailles feuilles : $d = 3$; $G = 32^3$; $f = 2$; $Niv_{max} = 14$

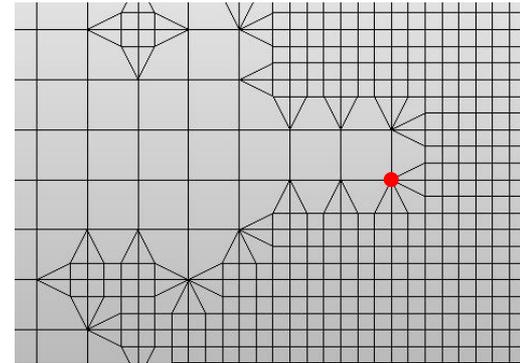


vtkHyperTreeGrid hérite de *vtkDataSet* (ens. de mailles et de points).

L'appel à certaines méthodes de *vtkHyperTreeGrid* (*getCell()*, *getPoint()*, etc) déclenche la création du maillage dual (complet).



Le maillage **primal** AMR (valeur au centre de la maille)



Le maillage NS **dual** (valeur au nœud)

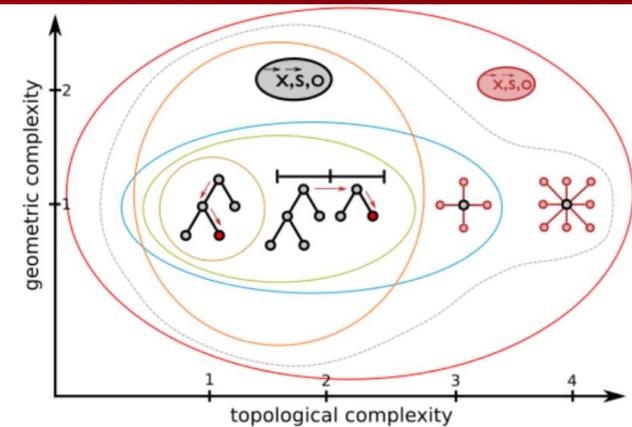


nativement, tous les filtres nécessitant un *vtkDataSet* en entrée fonctionnent comme les filtres de rendu, iso-contour, etc.



occupation mémoire élevée

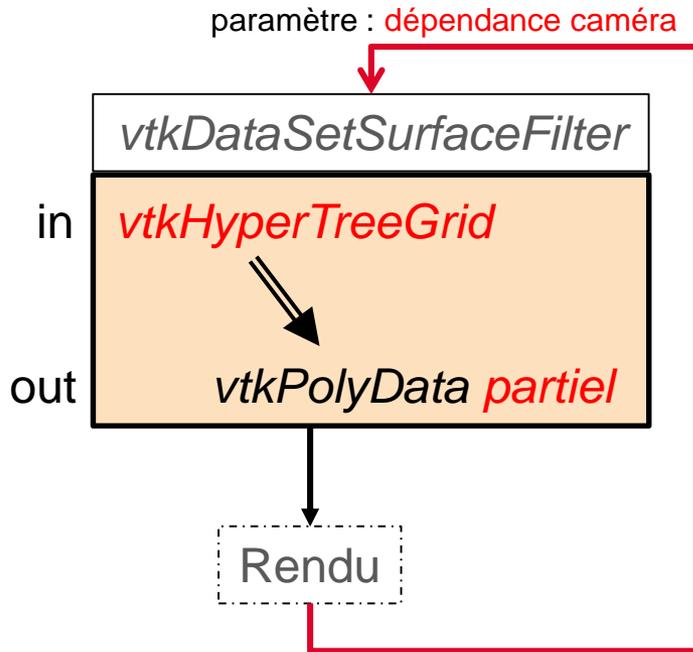
- 5 super-curseurs / itérateurs :
parcours optimisés des arbres sans création du dual



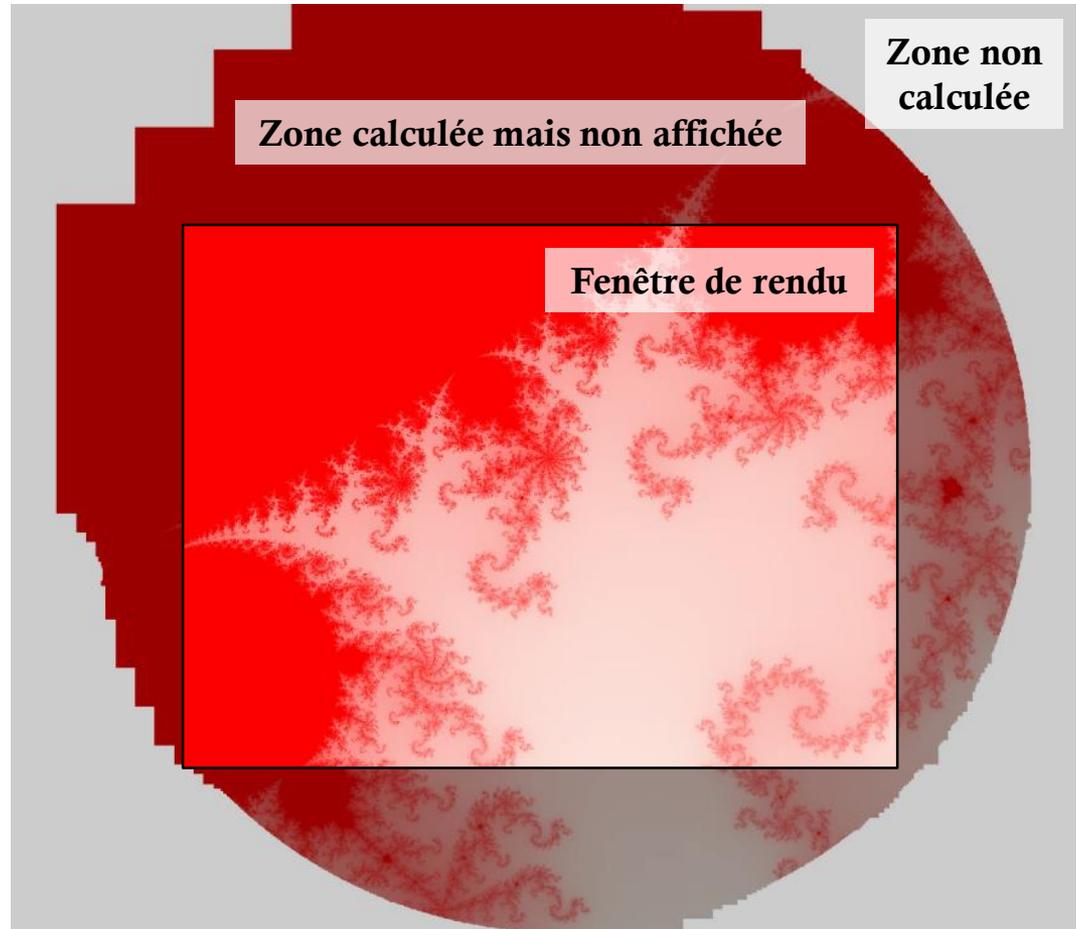
- Filtres spécifiques, optimisés pour l'AMR, utilisant un à deux super-curseurs

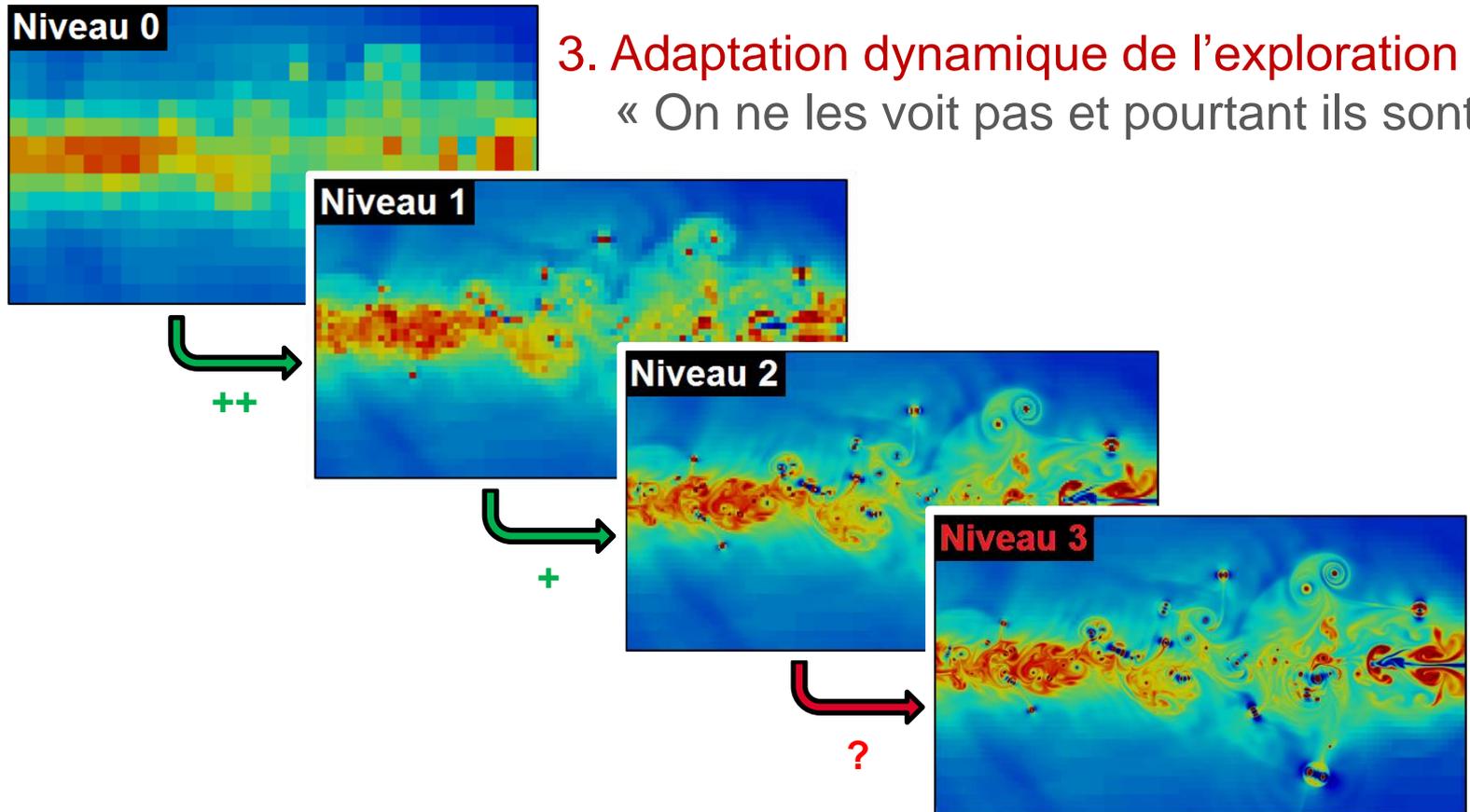
	TreeCursor	TreeGridCursor	GeometricCursor	VonNeumannSuperCursor	MooreSuperCursor
AxisClip	✓(output)		✓		
AxisCut	✓(output)		✓		
AxisReflection					
CellCenters			✓		
Contour		✓(pre-processing)			✓
DepthLimiter	✓(output)	✓			
Dual					✓†
ExtractSelected*			✓		
Geometry			✓($d < 3$)	✓($d = 3$)	
PlaneCutter primal			✓		
PlaneCutter dual		✓(pre-processing)			✓
Threshold	✓(output)	✓			
ToUnstructuredGrid			✓		

1. Adaptation dynamique du rendu « Loin des yeux, loin du cœur. »



2. Agrégation par couleur « Noir, c'est noir. »

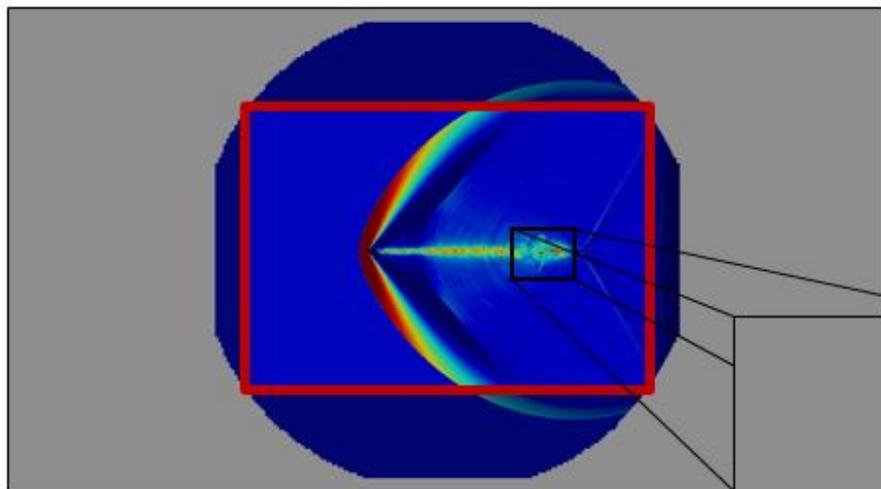




3. Adaptation dynamique de l'exploration
« On ne les voit pas et pourtant ils sont là. »

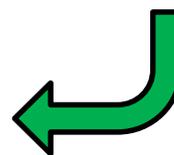
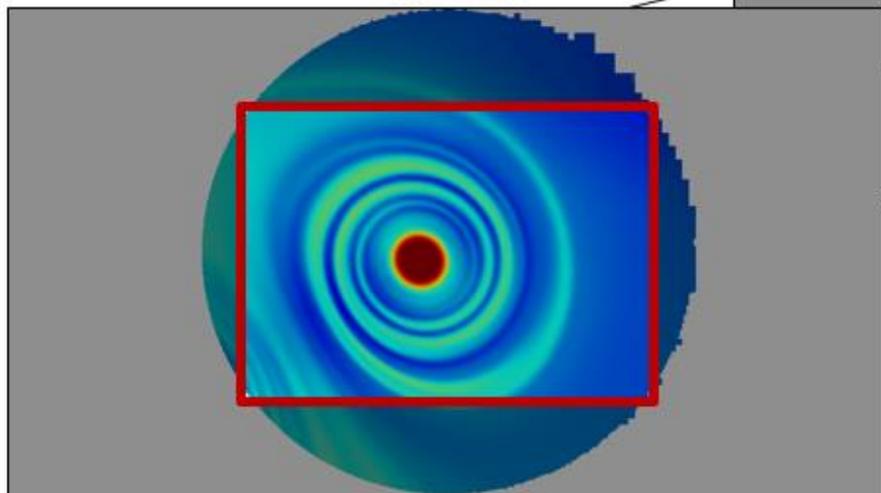
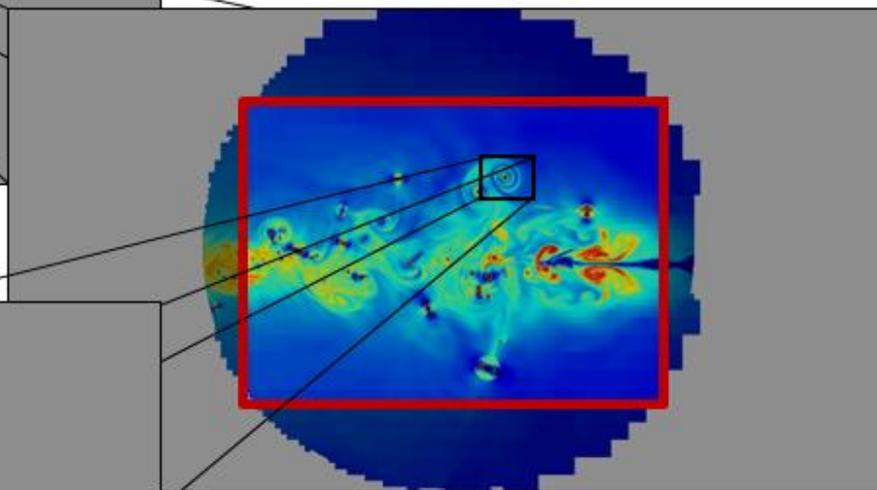
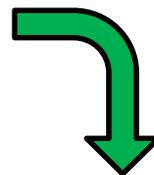
Limitation de l'exploration en fonction de la caméra et de la fenêtre de rendu :
contrôle de la valeur affichée de la maille.

FILTRE SPECIFIQUE DE RENDU POUR L'AMR 2D (3)



➤ Cas 105 millions de mailles feuilles :

$$d = 2 ; G = 168^2 ; f = 3 ; \text{Niv}_{\max} = 6$$



FPS : 3 img./s (en séquentiel)

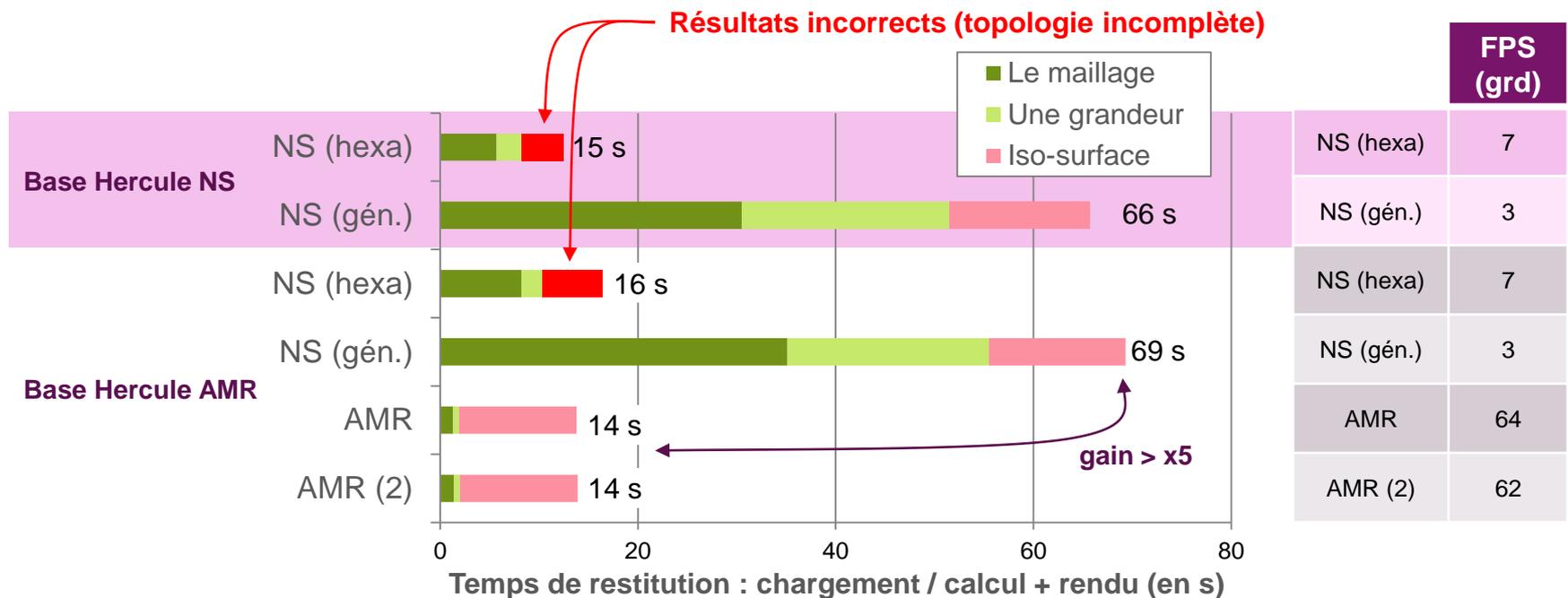
FPS LoD : 13 img./s

(3 niv. de moins dans vtkDataSetSurfaceFilter)



OPTIMISATION MÉMOIRE ET PERFORMANCE

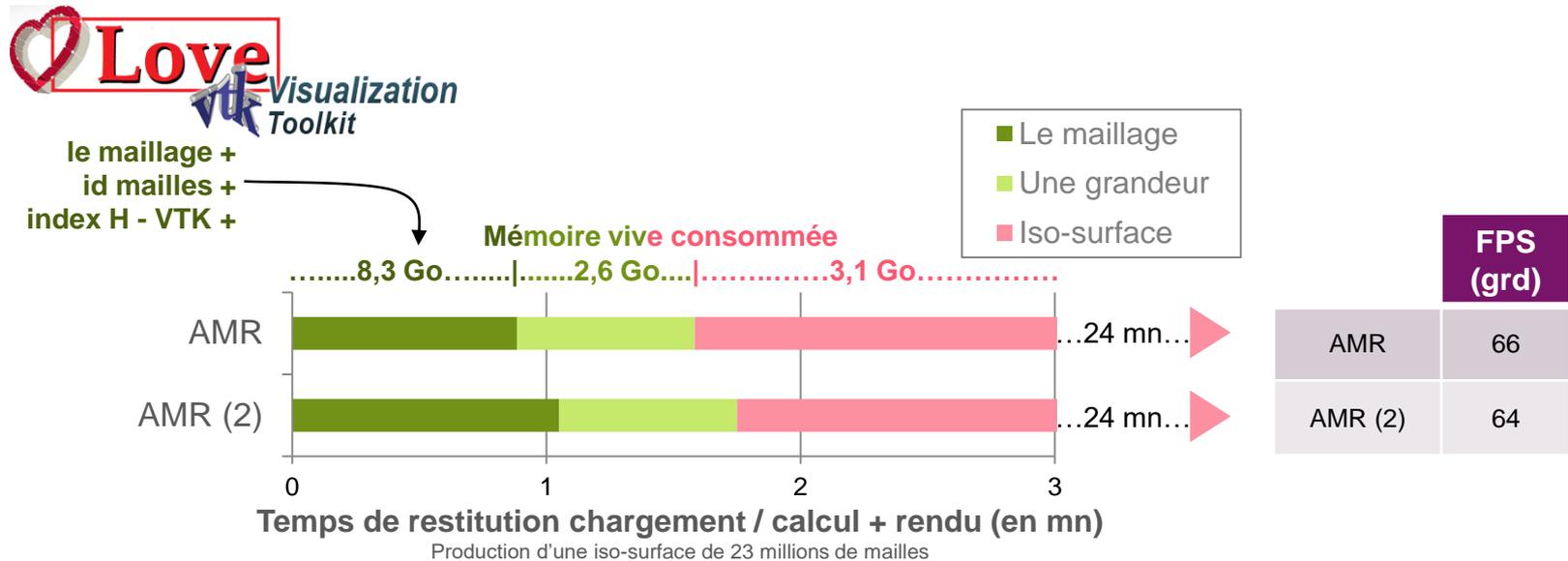
- **Cas 4 millions de mailles feuilles** : $d = 3$; $G = 32^3$; $f = 2$; $Niv_{max} = 14$
 Temps de restitution (séquentiel, station de travail 16 Go de mémoire)



Mémoire divisée par 50
Temps de restitution divisé par 5

300 MILLIONS DE MAILLES SUR UNE STATION

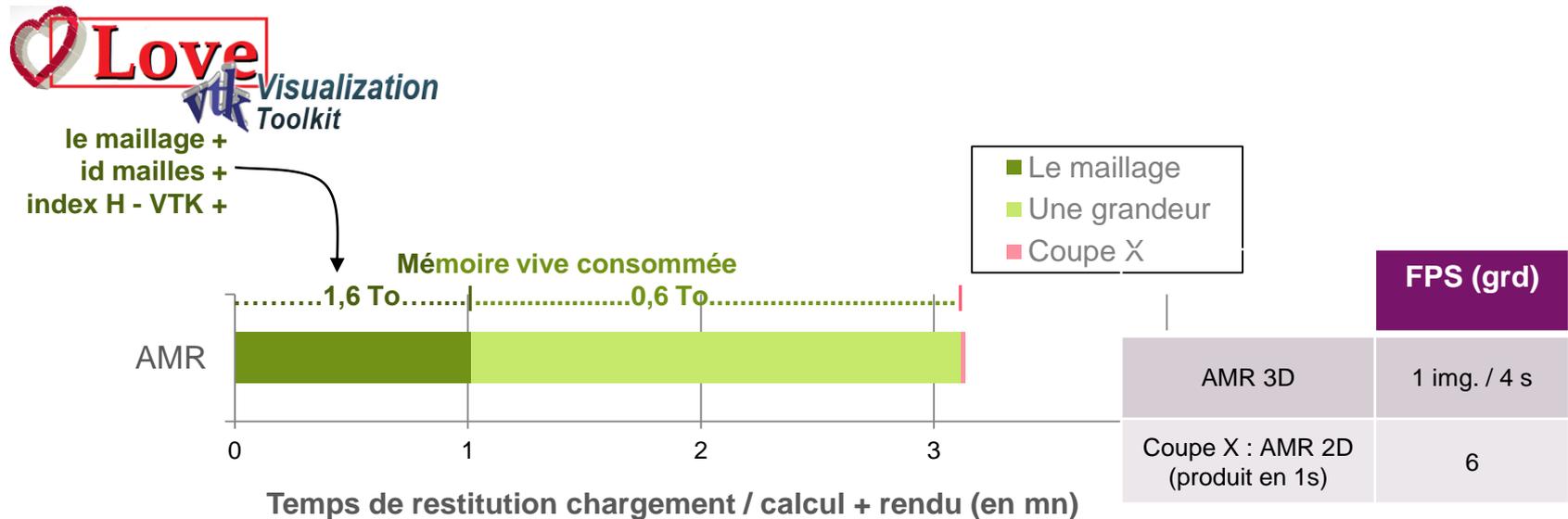
- Cas 286 millions de mailles feuilles : $d = 3$; $G = 128^3$; $f = 2$; $Niv_{max} = 13$
 Temps de restitution (séquentiel, station de travail 16 Go de mémoire)



Calcul cosmologique « pure matière noire » sur 4096 cœurs (CEA - Irfu, Ramses)

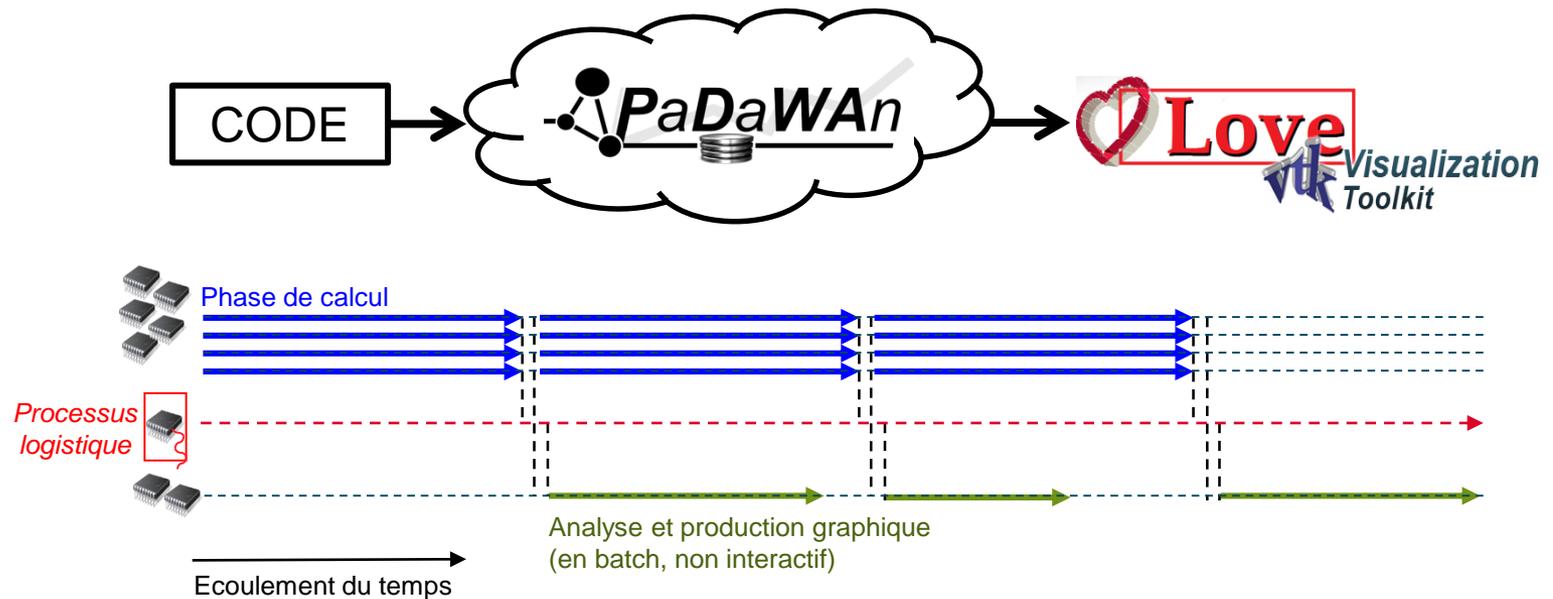
➤ Cas 72 milliards de mailles feuilles : $d = 3$; $G = 64^3$; $f = 2$; $Niv_{max} = 13$

Visualisation parallèle avec 128 cœurs et 3 To de mémoire réservée.



70 milliards de mailles visualisées en consommant 2,3 To
Visualisation interactive (slicing) : FPS 1 img. / s
32 fois moins de ressources que le code

- L'optimisation des ressources (réduction du nombre de CPU) facilite la mise en place d'enchaînement au code de traitement graphique à la volée.
- L'enchaînement de codes



CONCLUSION ET PERSPECTIVES

- Structure compacte pour l'AMR Tree-based dans VTK
- Développement de filtres spécialisés et optimisés : (ultra) performants
- Visualisation interactive sur station de travail de « gros » maillages
- Gestion optimisée des ressources : augmente l'intérêt pour l'enchaînement de traitements graphiques en cours d'exécution du code

Nos prochains développements :

- Ecriture de rendus 3D sans production de maillage NS ;
- Diverses optimisations comme la spécialisation aux grilles uniformes ;
- Sérialisation de la structure : comm. interne (*ghost* et ré-équilibrage) et format stockage (VTK).

Mais surtout :

- Reversement imminent dans  ;
- Collaboration CEA-LANL pour des développements en commun.

Merci pour votre attention



guenole.harel@cea.fr
jacques-bernard.lekien@cea.fr
philippe.pebay@ng-analytics.com