

Techniques de réduction contrôlée pour accélérer l'analyse de données de simulations massives dans VTK

Highly Efficient Controlled Hierarchical Data Reduction techniques for Interactive Visualization of Massive Simulation Data in VTK

Jérôme Dubois, Guénolé Harel et Jacques-Bernard Lekien

English Abstract—With the constant increase in compute power of supercomputers, high performance computing simulations are producing higher fidelity results and possibly massive amounts of data. To keep visualization of such results interactive, existing techniques such as Adaptive Mesh Refinement (AMR) can be of use. In particular, Tree-Based AMR methods (TB-AMR) are widespread in simulations and are becoming more present in general purpose visualization pipelines such as VTK. In this work, we show how TB-AMR data structures could lead to more efficient exploration of massive data sets in the Exascale era. By exploiting geometric and hierarchical properties, we introduce controlled hierarchical data reduction and optimized rendering for the visualization and storing of very large data on very constrained resources. These ideas are illustrated on million to billion-scale native TB-AMR or meshes resampled to TB-AMR, with the HyperTreeGrid object and associated filters we have recently optimized and made available in the Visualisation Toolkit (VTK) for use by the scientific community.

1 INTRODUCTION

Dans le domaine du calcul haute performance, plusieurs types de maillages peuvent être utilisés pour représenter les données issues des codes de simulation [5]. Les techniques Adaptive Mesh Refinement (AMR) offrent une balance intéressante entre l'occupation mémoire pour la description du maillage et celle des grandeurs de la simulation en raffinant près des détails, de manière semi-structurée. Ces maillages peuvent être entre autres block-based [2] ou tree-based [3]. Les représentations AMR tree-based (TB-AMR) permettent d'exploiter des données extrêmement larges, et rendent possible le rééchantillonage fidèle de données non-AMR existantes en contrôlant l'empreinte mémoire. Dans ce papier, nous décrivons rapidement différentes techniques utilisant les propriétés hiérarchiques de l'objet HyperTreeGrid (HTG - AMR en grille d'arbres) [6] [7] présent dans le Visu-alization Toolkit (VTK) [9] pour optimiser les temps d'analyse: réduction fidèle de maillage dynamique pour la prévisualisation de résultats de traitements, réduction contrôlée du stockage des maillages, et rendu interactif optimisé de maillages 2D de très grande taille. Ces techniques ont été implémentées et expérimentées dans VTK. Cette approche permet de grands gains en temps pour les analystes, une haute

efficacité sur de grandes ressources de calcul, et la visualisation de jeux de données de classe Petaflopique sur un portable.

2 ACCÉLÉRATION DES ANALYSES

Durant la phase exploratoire, le filtrage des données peut être accéléré en limitant la profondeur des arbres. Nous avons implémenté un filtre de limitation de niveau pour l'HTG dans VTK. L'application de ce dernier est gratuite car une copie superficielle (shallow copy) de l'HTG a lieu et un scalaire est positionné pour indiquer le niveau maximum. Des gains considérables sont obtenus, permettant notamment un rendu volumique temps réel de grands jeux de données comme dans [4]. On peut également tester plusieurs fois par seconde des isovaleurs différentes pour le calcul d'isocontour. De manière générale, cette approche permet ainsi d'essayer plus de valeurs pour les filtres interactifs et également plus de points de vue. À noter, la structure HTG garde la description complète du maillage, et ainsi la consommation mémoire n'est pas impactée par le processus de limitation de niveau. Seules les données produites à l'issue de la limitation seront réduites en mémoire.

3 RÉDUCTION CONTRÔLÉE DU STOCKAGE

Cette limitation de niveau peut être appliquée pour les étapes de lecture et d'écriture sur disque. Cela permet une réduction contrôlée de la taille des données, qui peut être d'intérêt pour par exemple générer des prévisualisations rapides des données ou encore faire rentrer les données sur un matériel plus réduit.

- Jérôme Dubois: CEA, DAM, DIF, F-91297 Arpajon, France
E-mail: jerome.dubois@cea.fr.
- Guénolé Harel: CEA, DAM, DIF, F-91297 Arpajon, France
E-mail: guenole.harel@cea.fr.
- Jacques-Bernard Lekien: CEA, DAM, DIF, F-91297 Arpajon, France
E-mail: jacques-bernard.lekien@cea.fr.

A la lecture les IOs seront drastiquement réduits, et les étapes de filtrages des données ultérieures seront aussi accélérées. La consommation mémoire sera également réduite car le maillage HTG généré est lui-même limité en niveau. Dans un autre cas, si le maillage chargé est pleine résolution, et que les données ont été filtrées, alors il reste possible d'appliquer la limitation de niveau à l'écriture, permettant une accélération future lors de la relecture des données et de leur traitement ultérieur.

4 RENDU HIÉRARCHIQUE OPTIMISÉ

Le rendu peut aussi bénéficier de la géométrie exposée par l'HTG ainsi que des niveaux de raffinement de chaque arbre. En utilisant les informations de caméra et de résolution d'écran, nous pouvons considérablement accélérer le rendu de larges scènes. La connaissance de la caméra permet de ne sélectionner que les arbres concernés par le rendu, alors que la résolution permet de calculer le niveau maximum réellement affichable, évitant ainsi le parcours des sous-arbres correspondant aux informations sous-pixel. De plus, un sous-arbre affichable peut avoir la même valeur et dans ce cas, nous ne générerons qu'une seule maille de même couleur, optimisant l'occupation mémoire sur la carte graphique. Dans la section résultats, les framerates sont 10x à 1000x meilleurs et la consommation mémoire réduite d'un facteur 13x.

5 RÉSULTATS

Nous évaluons les techniques proposées avec les données suivantes, présentées sur la figure 1: (1) Shock100, simulation 2D TB-AMR de 10^8 mailles; (2) YA31, un maillage non-structuré 3D de 3.10^8 mailles [10]; (3) S1, un maillage structuré 3D de 10^9 mailles; (4) Mandel6, un maillage 2D TB-AMR de 6.10^9 mailles [8]; (5) Astro72, un maillage 3D TB-AMR de 72.10^9 mailles. YA31 and S1 sont des données rééchantillonnées en HTG. Dans ce papier nous ne développerons pas les techniques utilisées lors du rééchantillonage, mais la résolution complète des jeux de données d'origine a été maintenue. Astro72 est un jeu de données de très grande taille, demandant des Tera-octets de mémoire lors d'analyses complexes. Shock100 et Mandel6 sont des maillages 2D illustrant le rendu optimisé avec l'HTG. Les tests ont eu lieu sur soit un portable Dell Latitude 7480, Core i5-6300U, 16 GigaOctets de mémoire et 512 GigaOctets de SSD, soit sur jusqu'à 16 nœuds du calculateur Tera-1000-1 [1], avec 2* Xeon E5-2698v3 16C 2.3GHz, 128 GigaOctets de mémoire par nœud et du stockage Lustre. Nous utilisons le nombre de nœuds nécessaires selon la taille de chaque jeu de donnée. Pour Astro72, nous utilisons 16 nœuds de Tera-1000-1, et moins de nœuds pour les autres jeux de données ou encore ceux que nous réduisons, voir uniquement un cœur de portable.

5.1 Filtrage accéléré

Dans le tableau 2 les temps de filtrage sont réduits de plusieurs minutes à moins d'une seconde sur le même pas de temps de YA31. Les images correspondant au premiers niveaux sont présentées dans les figures 2 et 3. Le temps de chargement initial prends 30 secondes et est indépendant de la limitation de niveau appliquée ultérieurement.

5.2 Chargement accéléré

Les temps de chargement des jeux de données limités en niveau sont présentés dans le tableau 1. Les gains de temps permettent d'identifier plus rapidement des temps d'intérêt dans la simulation. Pour YA31, l'astéroïde frappe la surface de l'eau après une dizaine de pas de temps. Avec le chargement accéléré, cela peut être trouvé en quelques secondes là où il faut plusieurs minutes normalement en résolution complète. L'eau atteint sa hauteur maximale après plusieurs centaines de pas de temps. On peut trouver cet instant en quelques minutes quasi-interactivement avec notre approche. Les gains obtenus sont directement liés au nombre de mailles ajoutés par chaque niveau, et pour Astro72, le dernier niveau ajoute peu de mailles sur le pas de temps testé, d'où un gain de stockage réduit. Avec la limitation de niveau du stockage, nous pouvons contrôler facilement la résolution d'Astro72 et rendre ce dernier exploitable sur un portable pour des besoins collaboratifs.

5.3 Rendu 2D optimisé

Le tableau 3 montrent la performance du rendu optimisé. Pour Shock100, l'interactivité passe d'une manipulation poussive à une expérience extrêmement fluide. Mandel6 n'est exploitable qu'avec notre approche, qui par ailleurs réduit drastiquement la consommation mémoire du GPU. Comme le rendu 2D optimisé tient compte de la résolution de l'écran, la performance est liée à la configuration de l'affichage: réduire la taille de la fenêtre d'affichage augmente linéairement la performance. L'optimisation de notre rendu libère de la mémoire, qui peut alors être utilisée par des pipelines GP-GPU.

6 CONCLUSION

Dans ce papier, nous avons rapidement présenté des techniques utilisant la structure TB-AMR HTG présente dans VTK pour permettre les rendus interactifs accélérés de données massives. En utilisant l'HTG présent dans VTK, notre limitation de niveau et notre rendu optimisé, les analyses peuvent être drastiquement accélérées. Nous pensons que ces techniques ouvrent la voie à des stratégies pérennes pour la visualisation de données à l'échelle Exaflopique, qu'elles soient originellement TB-AMR ou rééchantillonnée en TB-AMR pour le besoin de la visualisation.

REFERENCES

- [1] *Tera-1000-1*, november 2017. <https://www.top500.org/system/178790>.
- [2] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82(1):64 – 84, 1989.
- [3] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484 – 512, 1984.
- [4] J. Dubois, G. Harel, and J.-B. Lekien. Interactive visualization and analysis of high resolution hpc simulation data on a laptop with vtk. IEEE VIS'18 SciVis Contest, 2018.
- [5] P. Frey and P.-L. George. *Mesh generation*. John Wiley & Sons, 2 edition, 2008.
- [6] G. Harel, J. Lekien, and P. P. Pébay. Two new contributions to the visualization of AMR grids: I. interactive rendering of extreme-scale 2-dimensional grids II. novel selection filters in arbitrary dimension. *CoRR*, abs/1703.00212, 2017.
- [7] G. Harel, J. Lekien, and P. P. Pébay. Visualization and analysis of large-scale, tree-based, adaptive mesh refinement simulations with arbitrary rectilinear geometry. *CoRR*, abs/1702.04852, 2017.
- [8] P. Holmes. An introduction to chaotic dynamical systems (robert l. devaney); nonlinear dynamics and chaos: Geometrical methods for engineers and scientists (j. m. t. thompson and h. b. stewart) (with the assistance of r. ghaffari and c. franciosi and a contribution by h. l. swinney). *SIAM Review*, 29(4):654–658, 1987.
- [9] J. C. Moore. Visualizing with vtk. *Linux J.*, 1998(53es), Sept. 1998.
- [10] J. M. Patchett and G. Ross. Deep water impact ensemble data set. 2017.

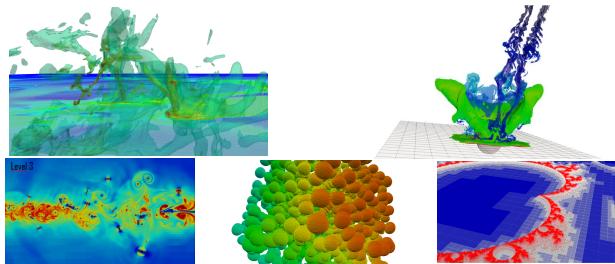


Fig. 1. Utilisation de l'HyperTreeGrid de VTK pour le rendu de différents jeux de données, en appliquant nos techniques de réduction contrôlée et de rendu optimisé. Sur la première ligne, le rendu de Astro72 sur 16 nœuds de Tera-1000-1 et le rendu d'YA31 sur 1 cœur de portable. Sur la seconde ligne, le rendu de Shock100 sur un cœur de portable, celui de S1 puis Mandel6 respectivement sur 1 et 2 nœuds de Tera-1000-1.

TABLE 1

Chargement en secondes et stockage en limitant le niveau des arbres. Le matériel utilisé est fixe par jeu de donnée.

Chaque colonne montre un intervalle de limitation de niveau: les niveaux 0 à 3, puis 4 à l'avant dernier niveau, et enfin le niveau maximum du jeu de donnée. Les lignes correspondent respectivement aux couples (jeu de donnée ; niveau max) suivants: (Shock100 ; 7), (S1 ; 7), (Mandel6 ; 10), (Astro72 ; 12) et (YA31 ; 6).

L0-3	L4-L _{max-1}	L _{max}
<0.5 ; <5MB	<1 ; 32-22TMB	2 ; 1.3GB
<1 ; <40MB	<10 ; 0.2-8.8GB	<60 ; 67GB
4-7 ; <100MB	7-20 ; 227MB-54GB	30 ; 144GB
<6 ; <2.4GB	9-240 ; 20GB-2TB	240 ; 2TB
<1 ; <100MB	<5 ; 405MB-1.5GB	20 ; 6.9GB

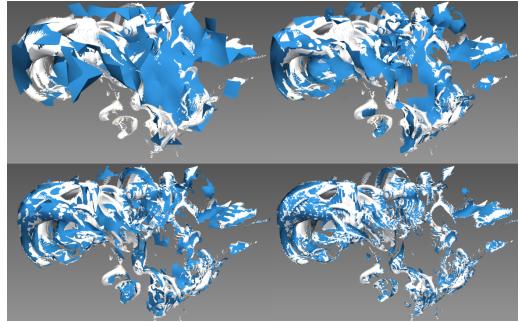


Fig. 2. Limitation du niveau avant isocontour de l'eau pour YA31. De gauche à droite, haut en bas, l'isocontour des niveaux 0 à 3 est montré en bleu. L'isocontour blanc est la valeur "exacte" en utilisant le niveau maximum. La limitation au niveau 4 (non montrée ici) ne montre aucune différence par rapport à l'isocontour blanc, de ce point de vue.

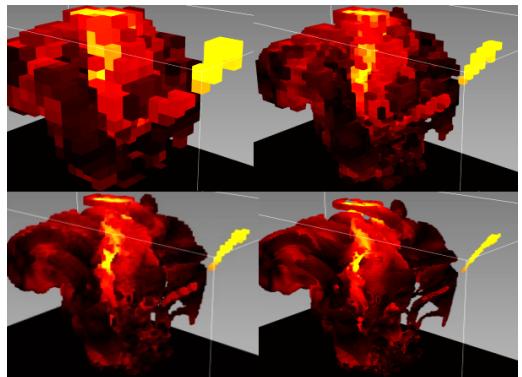


Fig. 3. Limitation du niveau avant threshold de l'élévation de l'eau d'YA31 aux niveaux 0 à 3, de gauche à droite, haut en bas. Jusqu'au niveau 2, les calculs de threshold sont instantanés.

TABLE 2

Temps d'exécution en secondes de filtres d'isocontour, threshold et rendu volumique après limitation de niveau pour YA31 au même pas de temps. L'accélération est directement corrélée à la réduction du nombre de mailles.

Depth limit	0	1	2	3	4	5	6
Contour	0.1	0.3	1.7	4	13	30	117
Thresh.	0.1	0.2	0.5	2	9	25	105
Volume	<1	1.5	2	7	20	64	285
Cells (10e6)	0.008	0.75	0.6	5	18	64	306

TABLE 3

Framerates et occupation mémoire du rendu 2D optimisé et naïf. La résolution de la fenêtre est de 3440x1440.

	Optimized Rendering	Standard Rendering
Shock100	8-20 fps	1 GB
Mandel6	1-5 fps	3 GB