REPORT ON (TOMS-2013-0068): DESIGN AND IMPLEMENTATION OF THE THREE-DIMENSIONAL AUTONOMOUS LEAVES GRAPH DATA STRUCTURE, BY: M.D. COSTA ET. AL

1. General issues and recommendation

This paper reports the 3D implementation of formerly invented method by the current authors in (Burgarelli et al, 2006), which is a graph based data representation corresponding to three dimensional spatial data structure on a 3D Cartesian grid with possibility of dynamic (graded) octree refinement. The main achievement of this approach is O(1) access to the data as well as the modification of data due to the dynamic data refinement (or coarsening). The method has potential applications in the solution of PDEs with adaptive grid refinement (more for FVM, cell, based approaches), deformable complex body modeling, etc. The computer implementation of the presented algorithm is provided by the authors as the supplementary material.

In general the topic is interesting to a broad range of computational science community. However, I believe that this paper is not suitable for publication in ACM-TOMS due to several reasons. 1) The original contribution is little (O(1) algorithms are recently very mature, and this paper differs with conventional methods mainly for introducing transition nodes in graph to connect cells with different refinement level), 2) The existence of many O(1) and efficient algorithms for this purpose with available open-source implementations (in particular with support for massive computation using parallel processing technique) and serious ambiguity about whether the approach of the current author is superior than them? there is no comparison, neither theoretically nor computationally (in my opinion, based on my immediate judge, it has no preference). 3) The style of presentation is very poor, literature survey is quite blind about available alternative approaches and the place of this method within available approaches is not clear. Moreover, the paper is unnecessary lengthy and verbose. 4) Numerical section also give us no promise about the method, no quantitative report on computational cost, proof of complexity, scalability of the computational cost as a function of total number of cells and level of refinement, no report on the solution of challenging problems which need massive computation; but some results like some snapshots.

In summary, I could not recommend this paper for the publication in ACM-TOMS. In the following is my comments to authors for improving their work.

2. Major commnets:

1) **Literature survey and motivation:** The literature survey is very poor. No alternative method in this regard is considered by the authors, consequently no comparison and illustration for the preference of the authors' approach in contrast to them is reported. Here is only a few instances of available approaches: Efficient generation of octree refined mesh (and data) and its corresponding data structure was discussed in details many in (Aftosmis et al, 1998) (it has been implemented in the Cart3D code). The parallel implementation of this method by the domain decomposition approach and increasing data locality by arrangement of data based on the Hilbert space filing curves were discussed in (Berger et al, 2005). Comparison with reverse Cuthill-McKee data reordering, i.e., using essentially unstructured data in contrast to semi-structured data (octree), which is another way to increase the data locality (to reduce the cache-miss) has been discussed in (Aftosmis et al, 2000) (implies almost the same performance, some reference also report superior performance due

to further increasing the cache coherency). Fully threaded tree (FTT) algorithm to parallel management of quadtree-octree spatial data in O(1) time has been introduced in (Khokhlov, 1998) (see also (Kravtsov et al, 1997)). It could be easily extended to N-d situations. FTT data structure used in (Popinet, 2003) to develop Gerris flow solver code which is a an open-source code to solve PDEs on 3D (2D) dynamically adapted Cartesian octree (quadtree) grids. This code is recently very mature and has been used extensively in scientific computing community to solve PDEs, in particular solving real world application needs very fine (effective) grid resolutions. The full implementation of FTT data structure is available in an OOP-like fashion in ftt class of this code. Hierarchical Run-Length Encoded (H-RLE) data structure, with O(1) data access complexity, for octree representation of complex data with dynamic adaptation support is introduced in (Houston et al, 2006). This method has been used successfully for applications with over 3000^3 effective grid resolution on a conventional PC. Ordering of adaptive spatial data based on the Hilbert space filling curve, to increase the data locality to reduce the cache miss, has been discussed in (Günther et al, 2006). Peano (Bungartz et al, 2010; Weinzierl and Mehl, 2011) provides a framework for octree data representation and management with dynamic adaptation. Parallel balance of octree data has been discussed in (Sundar et al, 2008) (the corresponding computer code has been also freely distributed by its authors). P4est (Burstedde et al, 2011) also performs the parallel management of complex data in a manner that the spatial domain is composed from several octrees (the combination of these octress are not essentially form an octree structure). Another kind of dynamic adaptive spatial data structure with non-square (cubic) elements has been introduced in (Strain, 2001) (in 2D) and (Bargteil et al, 2006) (in 3D). Another related work is (Bangerth et al, 2011) in which adaptive data structure (more suitable for nodal data) has been discussed. There are much more work in this regard, not mentioned here. I think that authors should do a comprehensive literature survey to determine the location of their method, then, to compare their method to available ones, to illustrate (demonstrate) its positive and negative points. Then, it is possible to decide on the novelty and importance of this work for publication in ACM-TOMS.

2) **The style of presentation:** The paper is unnecessarily lengthly (considering its relatively little original contribution). The organization of paper is also poor. It is divided to many sections and subsections which not only help the reader to easily understand the method, but increase the complexity and ambiguity. I think that its size could be significantly reduced in a revision. In introduction authors said that there are some "new set of interesting phenomena which can not be fully understood in a two dimensional spatial setting". Because 2D algorithm has been already presented in another works of these authors (see (Burgarelli et al, 2006)), I expect that authors focus only on the mentioned "phenomena" which make the implementation (possibility) significantly different (in fact, make it clear what is their original contribution). On the other hand, I think that there is no serious problem in the extension of 2D version of their algorithm to 3D (else I missed something). Moreover, I think that the presentation of basic idea will be more effective and brief if the authors describe their algorithm firstly in 2D. Then, it will be easy for reader to think about the 3D extension and authors could warn the readers about possible challenges and their solution in this regard. There are several excess descriptions in the paper. Just as an instance: many aspects of the refinement algorithm is similar to de-refinement (coarsening) step, while the authors describe them in two subsection, both with details. For many simple and evident procedures, distinct algorithms have been presented. Unfortunately, unlike authors waste the space a lot during the presentation of the algorithm some aspects of the algorithm which are very important in practice did not explicitly discussed. for instance authors did not explicitly indicate that whether their refinement method is graded (having 2:1 balancing), i.e., neighbor cells differ at most by one level. This issue is not discussed in algorithms.

3) **Cell ordering based on the space filling curve:** In section 5 authors describe the ordering of cells based on the space filling Hilbert curve. This approach is quite known and there is not necessary to discuss it in details. No reference is given in this regard. It is while there are very

mature literature on this topic. It is worth to discus about the improvement in the data locality (reducing the cache-miss) using such an ordering scheme. On the other hand, in some literature (cf. (Aftosmis et al, 2000)) authors claimed that using fully unstructured data reordered based on the reverse Cuthill-McKee data reordering has comparable (or even superior) cost in contrast to the Hilbert curve based reordering.

4) **The parallel implementation:** The parallel implementation of the method make it more attractive in the computational science community; no sufficient discussion on this issue is presented in the paper.

6) **Presented results:** In section 6 authors used their algorithm for numerical solution of a PDE system corresponding to a simulation in electrophysiology. The details of numerical algorithm based on the FVM is discussed. However, there is no quantitative results on the performance of algorithm. I expect to use explicit solution algorithm and then report of the CPU time as a function of number of cells, number of refinement levels and difference between largest and finest level in practice. In fact there is no computational demonstration on the O(1) complexity of the algorithm. For this, it is essential to show that the computational cost of an explicit time integration step scales linearly by the number of cells. Moreover, it is required to report the memory usage as a function of number of cells. Then it is worth to compare these factors to at least an alternative method to demonstrate preference of the presented method. For instance this kind of problem could be easily solved by Gerris Flow Solver. Another interesting issue is reporting the computational cost for very large scale problem using a multiprocessor machine.

7) **The English writing:** I think that the style of English writing is not good. There are several grammatical mistakes as well as bad usage of language and words. Moreover, there are many lengthy statements which make it difficult to understand the goal of the author. I recommend a careful proof reading to fix such issues. In the following are a few instances of some obvious grammatical mistakes (There are several problems in structure of sentences, are so much, I was not so patient to list them here):
7.1: abstract: detail → details
7.2: Page 1, line 3: require → requires
7.3: Page 2, line 6: change → changes
7.4: Page 4, line 6: its → their
7.5: Page 8, line 3: some physical variable → some physical variables
7.6: Page 11, line 11: these → they
7.7: Page 14, line 3: who → that
7.8: Page 18, line 25: Therefore is → Therefore it is
7.9: Page 21, line 14: some direction → some directions
7.10: Page 25, line 6: ... is that ... → ... is the one that ...
7.11: There are some unnecessarily lengthy sentences, for instance see second sentence if the first paragraph of page 8.

8) **Bibliography:** Some bibliography items are incomplete, e.g. (based on the presented order) item 1 and 7.

9) **Supplement (computer code):** The computer implementation of the presented algorithm is provided by the authors as the supplementary material to this paper. The code structure and names of variables and classes are identical to those of the paper. The level of comment in the code is good and sufficient to follow it. However, some issues are missed: 1) There is no user guide about minimum pre-requirement to compile the code (OS, required library, etc), 2) There is no auto-configuration tool, 3) There is no test folder to evaluate the basic features of the algorithm, but a demo application. I think that it is difficult for the reader to use this code for own application

without considerable efforts.

References

Aftosmis MJ, Berger MJ, Melton JE (1998) Robust and efficient cartesian mesh generation for component-based geometry. AIAA journal 36(6):952–960

Aftosmis MJ, Berger MJ, Adomavicius G (2000) A parallel multilevel method for adaptively refined cartesian grids with embedded boundaries. AIAA paper 808:2000

Bangerth W, Burstedde C, Heister T, Kronbichler M (2011) Algorithms and data structures for massively parallel generic adaptive finite element codes. ACM Transactions on Mathematical Software (TOMS) 38(2):14

Bargteil AW, Goktekin TG, O'brien JF, Strain JA (2006) A semi-lagrangian contouring method for fluid simulation. ACM Transactions on Graphics (TOG) 25(1):19–38

Berger MJ, Aftosmis MJ, Marshall D, Murman SM (2005) Performance of a new cfd flow solver using a hybrid programming paradigm. Journal of Parallel and Distributed Computing 65(4):414–423

Bungartz HJ, Mehl M, Neckel T, Weinzierl T (2010) The pde framework peano applied to fluid dynamics: an efficient implementation of a parallel multiscale fluid dynamics solver on octree-like adaptive cartesian grids. Computational Mechanics 46(1):103–114

Burgarelli D, Kischinhevsky M, Biezuner RJ (2006) A new adaptive mesh refinement strategy for numerically solving evolutionary pde's. Journal of Computational and Applied Mathematics 196(1):115–131

Burstedde C, Wilcox LC, Ghattas O (2011) p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. SIAM Journal on Scientific Computing 33(3):1103–1133

Günther F, Mehl M, Pögl M, Zenger C (2006) A cache-aware algorithm for pdes on hierarchical data structures based on space-filling curves. SIAM Journal on Scientific Computing 28(5):1634–1650

Houston B, Nielsen MB, Batty C, Nilsson O, Museth K (2006) Hierarchical rle level set: A compact and versatile deformable surface representation. ACM Transactions on Graphics (TOG) 25(1):151–175

Khokhlov AM (1998) Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations. Journal of Computational Physics 143(2):519–543

Kravtsov AV, Klypin AA, Khokhlov AM (1997) Adaptive refinement tree: a new high-resolution n-body code for cosmological simulations. The Astrophysical Journal Supplement Series 111(1):73

Popinet S (2003) Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. Journal of Computational Physics 190(2):572–600

Strain J (2001) A fast semi-lagrangian contouring method for moving interfaces. Journal of Computational Physics 170(1):373–394

Sundar H, Sampath RS, Biros G (2008) Bottom-up construction and 2: 1 balance refinement of linear octrees in parallel. SIAM Journal on Scientific Computing 30(5):2675–2708

Weinzierl T, Mehl M (2011) Peano-a traversal and storage scheme for octree-like adaptive cartesian multiscale grids. SIAM Journal on Scientific Computing 33(5):2732–2760