Acta Technica *62* (2017), No. 5B, 993–1002

Study of parallel delaunay triangulation using many-core processor

Zhang Jinzhu[1], Song Qingzeng[2,4], He Hua[1], Zhang Minglu[3]

Abstract. With the increasing of the scale of science and engineering problems, the mesh generation is becoming the bottleneck of the performance of the finite element problem. In order to improve the performance of grid partition, in this paper the implementation and optimization of parallel Delaunay triangulation algorithm on Xeon Phi many-core processor is studied. Delaunay triangulation algorithm includes three key steps, including initial mesh generation, parallel flip and repair. The non-Delaunay, flip ability and non-collision detection are carried out in turn by supporting parallel operation such as flip, search and replace, and the parallel flip operation is performed on the detected surface. The experimental results show that, for most of the test sets, the performance of the parallel execution can be achieved at least 4 times more than that of the CGAL software package, and achieve the goal of performance acceleration.

Key words. Finite Element Method, Many-Core Processor, Delaunay Triangulation.

1. Introduction

Mesh generation is an important step in finite element analysis and one of the main performance bottlenecks in finite element applications. In the extreme case, the time of mesh subdivision may exceed the solution time of the finite element equation. With the advent of multi-core and all-nuclear era, some researchers have begun to design and implement a parallel grid partition algorithm on parallel computing devices, such as multi-core CPUs and many-core processors, thus accelerating the calculation of gridding. The hardware architecture of these parallel computing devices is fixed. The essence of parallel implementation is to efficiently map the char-

[1]Workshop 1 - 1.School of Science, Hebei University of Technology, Tinjin 300401, China
[2]Workshop 2 - Institute ofComputer Science and Software Engineering, Tianjin Polytechnic University, China
[3]Workshop 3 - School of Mechanical Engineering, Hebei University of Technology, China
[4]Corresponding author: Song Qingzeng

acteristics of algorithms to parallel hardware architectures. The degree of coupling directly determines the performance of the actual computing. Therefore, parallel computing devices must be designed for the architecture of the appropriate parallel algorithms, data structures, in order to obtain better performance.

In three-dimensional space, Delaunay triangulation is the first choice of finite element analysis to establish the mesh, which can minimize the radius of tetrahedron, making the triangulation the most compact, so that the density of the mesh is the best. In the 1970s and 1980s, Delaunay triangulation was applied to the field of mesh generation, and showed some unique advantages, such as good theoretical basis, high quality grid and fast algorithm running. In the late 1980s and early 1990s, some Delaunay triangulations began to integrate into various grid generation software, and has been widely usedin the scientific research and engineering practice. Delaunay triangulation includes the key steps of automatic point set insertion, Delaunay triangulation and boundary restoration??which is the most time-consuming and the most critical step.

Since no formal papers have been published on parallel Delaunay triangulation based on Xeon Phi processors, the Delaunay triangulation algorithm on the Xeon Phi all-core processor is designed and implemented in this paper.

2. Relation Work

Parallel Delaunay triangulation generation algorithms can be summarized as "parallel strategy + traditional generation algorithm". Traditional generation algorithms mainly include Incremental Insertion Algorithm and Flipping Algorithm. Parallel strategy is mainly divided into D & C, parallel flip and parallel insert. There are mainly three kinds of the parallel devices such as multi-core CPU, GPU and Xeon Phi three.

Kohoutetal.[1]designed a two-dimensional and three-dimensional parallel Delaunay algorithm for multi-core CPU,which is based on the serial incremental interpolation method and use a DAG map to maintain a triangular grid, so that multiple threads can be inserted into a new point. Batista et al. [2]extended CGAL's support for multi-threading,which used a parallel strategy similar to that proposed by Kohout et al. [1], except that tetrahedral cell data structures and Bowyer-Watson methods were used for point insertion. Foteinosetal.[3] used vertex-locked parallel strategyto achieve a dynamic Delaunay triangulation algorithmin a multi-core CPU, which, to a certain extent, affects its acceleration effectbecause of supporting linear point set insertion and deletion operations.

Hoff[4] proposed the use of graphics cards for discrete two-dimensional and three-dimensional Voronoi diagram, and proposed the possibility of using discrete Voronoi diagram for Delaunay triangulation, but failed in realization. Rong[5] proposed a two-dimensional Delaunay triangulation algorithm based on GPU-CPU, which can obtain 1.53 times speedup for the uniformly distributed point set. Qi[6]designed the GPU-CDP algorithm on the base of the GPU-DT algorithm to complete all calculations without CPU participation, which achieved 4.5 times speedup. Gao et al.[7] designed the GHull algorithm for calculating three-dimensional convex hulls,which

can obtain about 10 times performance speedup on GPU.

3. Algorithm Design

3.1. Algorithm Flow

Insertion and flip are the most widely used local operations in Delaunay triangulation. Joe[8] proved that, if we insert a point exclude a point set into a Delaunay triangulation, we will obtain a Delaunay triangulation of the new point set by performing the flip operation. This conclusion is the basis of constructing Delaunay triangulation by using many kinds of incremental interpolation methods. However, the incremental interpolation method needs to maintain the Delaunay grid, which is too costly and very difficult to parallelize. Thus, instead of trying to maintain a Delaunay grid, inserting multiple points once and then flipping can improve the quality of the mesh until no flip. Repeating the above steps until all the points are inserted into the mesh and the resulting mesh does not have a face that can be flipped. There are still some faces in the mesh which fall into the NLONT loop. To solve this problem, it needs to repair the approximate Delaunay triangular mesh into a real Delaunay triangular mesh by the repair algorithm, which is shown in Figure 1.

	Algorithm
1:	procedure OPT(S)
2:	while S ≠ ϕ do
3:	POINTINSERT (S, T, α)
4:	FLIP (T)
5:	end while REPAIR(T)
6:	return T
7:	end procedure

Fig. 1. Pseudocode of the parallel algorithm

3.2. Insert Operation

The insertion operation is divided into three stages: selecting the inserted point, inserting the point and reallocating the points. Select the points, which should be separated, to make the insertion operation and the subsequent flip operation easier to parallelize. However, the complexity of the operation will reduce the performance of the algorithm, for which we select the internal points of the existing tetrahedrons.

After selecting the point, the point is needed to be inserted into the corresponding tetrahedron. As the new points are inserted, the inserted tetrahedron will split into four new tetrahedra. Besides, we need to read the original tetrahedron and the insertion points to produce a new tetrahedron with a 1-to-4 flip. The point set in the original tetrahedron also need to be re-assigned to the four tetrahedron, which is named the point reallocation operation.

3.3. Flip Operation

The flip operation is a manipulation of the non-Delaunay and the flippable tetrahedron in the triangulation. The flip operation will occur several times until there is no non-Delaunay and flippable faces in the triangulation. Next it needs to determine whether a 2-to-3 or 3-to-2 flip operation would do under the case of convexity. If the flipping detection passes, a 2-to-3 or 3-to-2 flipping is performed. The flip operation is repeated until all faces meet the Delaunay property, or can not be flipped.

3.4. Repair Operation

The criterion for choosing the repair algorithm is that its runtime is only related to the number of non-Delaunay tetrahedra in the initial grid. The Star splaying algorithm in paper [9] satisfies this property. This paper uses Star splaying algorithm to repair the approximate Delaunay mesh. It should be noted that the data dependency of the repair algorithm is more complicated and basically difficult to be parallelized.

4. Parallelization Strategies

In order to achieve parallel flip on the Xeon Phi processor, parallel data structure is needed to support flip, find, replace and other operations. Besides a variety of detections are required for the flip operation in parallel. Certain distance between the points is needed forinserting multiple points in parallel.

4.1. Build the Data Structure for Parallelization

Efficiency is very important for the data structure for storing the tetrahedron mesh in three-dimensional space. On the other hand, it is necessary to consider whether reallocation, insertion and flip can be efficiently realized based on the data structure. In this paper, a tetrahedron cell is selected as a triangular storage structure, in which the triangular mesh is represented by an ordered sequence table. Each tetrahedron stores its indexes of four vertices and directions. In addition, each tetrahedron also stores its four adjacent tetrahedron indexes. If a point or tetrahedron is stored in a dynamically allocated structure, a pointer can be used instead of an index. Tetrahedrons in the tetrahedral cell format are represented by two nodes, one for the vertices of the tetrahedron stored in the directed list, and the other for adjacency of a tetrahedron. Therefore, this structure facilitates the execution of the

point insertion operation and the flip operation.

4.2. Parallel Flip Operation

When a surface satisfies both the local non-Delaunay property and the flippable property, the flip operation can be performed. However, due to the need of parallel, non-collision detection is also essential.

Local Delaunay property detection is the inner ball detection for the face. If the corresponding adjacent tetrahedron does not pass the ball detection, then it needs to find whether there is a flappable structure containing these two tetrahedrons. If not, then go on detecting the next adjacent tetrahedron.

The flappabledetection is to detect the presence of a 2-to-3 flip-flop structure or a 3-to-2 flip-flop structure. If the hexahedra with two tetrahedra bonded together are not convex, then they cannot be 2-to-3 flipped, which can be obtained by performing 1-3 direction detections. A 3-to-2 reversible structure is a polyhedron structure in which three tetrahedrons share a common edge.

Collision detection refers to whether other threads are operating simultaneously this flipable structure. If the values of all tetrahedra satisfy the smallest index for all tetrahedra in the flip-flop structure, flip can be done. Otherwise it means that at least one tetrahedron is still contained in another non-flippable structure. Then other structures can be preferentially flipped.

4.3. Parallel Insert Operation

Parallel insert operation is divided into two steps Selecting the insertion point and performing the insert. In the first step, select a point for each tetrahedron that contains an uninoculated point. In order to select a point for each tetrahedron in parallel, the insertion point method can be divided into calculating the distance for each point to be inserted and selecting the insertion point for each tetrahedron. If the distance from one point is found to match this value, mark it as the insertion point. After selecting the insertion point for each insertable tetrahedron, inserting the point into the corresponding tetrahedron. This step involves updating the information of the four new tetrahedron splitted by each insertable matrix and setting the internal and external adjacency information.

5. Experimental Results and Analysis

5.1. Experimental Environment

The paper uses a single-node CPU + Xeon Phi heterogeneous computer system for testing. Test system is the two-way Intel E5-2640 server, with 64GBmainmemory size, theRed hat 6.4 operating system,and the ICCcompiler. Xeon Phi card model SE7110p, the core clock frequency of 1.091GHz, the core of 61. Each core can run four threads. The Xeon Phi card has a memory size of 8GB and a memory type of GDDR5.

5.2. Test Set and Control Group

The five randomly distributions used to test the performance of the parallel algorithm are Uniform, Grid, Ball, Sphere and Gaussian. Evenly distributed points are composed of evenly generated points in a three-dimensional cube. The Gaussian is a distribution of points generated by a Gaussian function whose coordinates are in the range of[0.0,1.0]. The range of random points generated by the mesh distribution is[0,1024]. The Sphere is a point set evenly distributed within a sphere with a radius of 0.5. Sphere is evenly distributed on the surface of a sphere with a thickness of 0.05. As CGAL software package is the most widely used computational geometry library, in this paper, we compare and analysis the package execution time which the Delaunay triangulation algorithm cost in CGAL software.

5.3. Experimental Results and Analysis

Figure 2 shows the performance comparison between the parallel algorithm and the CGAL software package. It is concluded that the acceleration ratio of the parallel algorithms gradually increased with the problem increases, and the parallel algorithm is more suitable for dealing with large-scale problems. In addition, the parallel algorithm achieves the goal of performance acceleration when the size of the point set exceeds 600K, at least four times the speedup.

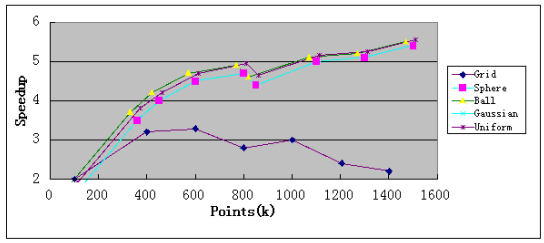

Fig. 2. Performance of the algorithm speedup

Table 1-3 show the ratio of the three steps of the parallel algorithm at different data scales. From the three tables, it can be concluded that the repair operation occupies a large proportion, and in particular, the proportion of repair operations did not decrease with the size of the data smaller and smaller.

Table 1. The proportion of 200 K points

Type	Mesh Generation	Parrallel Flip	Repair
Uniform	5.2%	75.4%	19.40%
Gaussian	4.6%	78.2%	17.20%
Ball	6.7%	77.8%	15.50%
Sphere	6.3%	73.3%	20.40%
Grid	7.8%	78.2%	14.00%

Table 2. The proportion of 400 K points

Type	Mesh Generation	Parrallel Flip	Repair
Uniform	2.2%	72.5%	25%
Gaussian	1.9%	76.2%	22%
Ball	2.4%	73.4%	24%
Sphere	2.3%	70.3%	27%
Grid	2.5%	75.2%	22%

Table 3. The proportion of 1400 K points

Type	Mesh Generation	Parrallel Flip	Repair
Uniform	0.93%	73.2%	26%
Gaussian	0.84%	74.2%	25%
Ball	0.87%	73.8%	25%
Sphere	0.96%	71.3%	28%
Grid	0.95%	74.2%	25%

Table 4 shows the ratio of the number of non-Delaunay tetrahedrons to the total set of points after the parallel flip operation. From the table, the ratio is still relatively large, at least 0.5%, and the proportion of the scale of the data also increases with the gradual increase. This leads to a large proportion of the repair steps in the total run time. Therefore, the number of non-Delaunay tetrahedrons can be reduced only after the parallel inversion is completed, so that the performance of the algorithm can be further improved.

Table 4. Percentage of non-Delaunay points to all point sets

types	200	400	600	800	1000	1400
Uniform	0.8%	0.9%	0.9%	1.0%	0.7%	0.9%
Gaussian	0.8%	0.5%	0.8%	1.2%	1.0%	1.3%
Ball	0.6%	0.7%	0.7%	0.8%	1.2%	1.2%
Sphere	0.9%	0.9%	0.8%	0.8%	1.5%	1.3%
Grid	0.7%	0.8%	0.5%	1.1%	1.1%	1.2%

6. Conclusions

In this paper, parallel Delaunay triangulation algorithm is designed and optimized for the hardware architecture of Xeon Phi many-core processor, comparing the performance of CGAL software package in different test set. The experiment results show that, for most of the test set, parallel execution can achieve at least 4 times more performance acceleration to achieve the purpose of performance acceleration. Due to the limitation of the condition, the algorithm designed in this paper can only run under the condition of single Xeon Phi accelerator card without using single multi-accelerator card or multi-machine accelerator card,whichlimit us to handle the data of the limited size (memory only 8GB), without the probability of handling larger problems. In the future we will further study the Delaunay triangulation algorithm for multi-accelerator card collaboration and in cluster implementation and optimization problems.

References

[1] J. Kohout, I. Kolingerova, J. Zara: *Parallel Delaunay triangulation in E^2 and E^3 for computers with shared memory.* Parallel Computing *31* (2005), No. 5, 491–522.

[2] H. F. V. Batista, D. L. Millman, S. Pion, J. Single: *Parallel geometric algorithms for multi-corecomputers.* Computational Geometry *43* (2010), No. 3, 663–677.

[3] A. P. Foteinos, A. N. Chernikov, N. P. Chrisochoides: *Fully Generalized Two-Dimensional ConstrainedDelaunay Mesh Refinement.* SIAM J. Sci. Comput *32* (1985), No. 5, 2659–2686.

[4] E. K. Hoff, J. Keyser, M. Lin, D. Manocha, T. Culver: *Fast computation of generalized Voronoi diagrams using graphics hardware.* In Proceedings of the 26thannual conference on Computer graphics and interactive techniques, 277–286.

[5] G. D. Rong, T. S. Tan, T. T. Cao: *Computing two-dimensional Delaunay triangulation using graphics hardware.* In Proceedings of the 2008 symposium on Interactive 3D graphics and games.

[6] M. Qi, T. T. Cao, T. S. Tan: *Computing 2D constrained Delaunay triangulation using the GPU.* In Proceedings of the ACM SIGGRAPH Symposium onInteractive 3D Graphics and Games *1.*

[7] M. C. Gao, T. T. Cao, A. Nanjappa, T. Tan: *A GPU Algorithm for Convex Hull.* ACM Transactions on Mathematical Software (2013).

[8] B. Joe: *Delaunay Versus Max-Min Solid Angle Triangulations for Three-Dimensional*

Mesh Generation. International Journal of Numerical Method in Engineering *31* (1991), No. 5, 987–997.

[9] A. Nanjappa: *3D Delaunay Triangulation on the GPU.* PhD Thesis (2012).

[10] J. Y. Yao, Z. X. Jiao, D. W. Ma, L. Yan: *High-accuracy tracking control of hydraulicrotary actuators with modelling uncertainties.* IEEE/ASME Transactions on-Mechatronics *19*, (2014), No. 2, 633–641.

Received November 16, 2017