Journal of Computational Science & Engineering 3 **(2012) 113-119**

Journal of Computational Science & Engineering

Available at http://www.asocse.org **ISSN 1710-4068**

Two Dimensional Parallel Delaunay Mesh Generations Based on Multi-core CPU Environment

Yu-hang Zeng [a], Hai-sheng Li [a], Qiang Cai[a], Yue-wu Liu [b*]

[a]*College of Computer and Information Engineering, Beijing Technology and Business University, Beijing100048, China*

[b]*Institute of Mechanics, Chinese Academy of Sciences, Beijing100190, China*

Article Information

Article history:
Received 16 June 2012
Revised 28 August 2012
Accepted 18 October 2012
Available online 28 October 2012

Keywords:
Parallel Delaunay Mesh
OpenMP
Irregular Region
Decomposition

Abstract

Based on the traditional parallel Delaunay mesh generation method, the sub-domain is mapped to the processor, but it does not consider the adjacency relationship among the sub-domains. Sub-mesh which is generated on a single processor usually can be composed by multiple of nonadjacent sub-domain mesh, and the number of the sub-mesh nodes shared among them is massive. According to the problem above, an improved parallel Delaunay mesh generation algorithm by using geometric domain decomposition strategy is proposed. The Delaunay mesh is generated by decomposing the complex two-dimensional geometric region into several sub-domains, using OpenMP technology to assign sub-domain dynamic to different processors, and calling Delaunay cavity algorithm for each sub-domain. Experimental results show that the proposed algorithm for the outer boundary of any irregular region, the inner boundary is a circular area can get better quality Delaunay triangular mesh.

1 Introduction

Delaunay mesh has a good theoretical basis and mathematical properties; it can simulate the irregular boundary and solve convergence problem. In addition, the mesh refinement and coarsening nature is better. Therefore Delaunay mesh has good application prospects in the field of engineering technology, especially in numerical reservoir simulation, computational electromagnetic, groundwater exploration, robot path planning, fluid

* To whom correspondence should be addressed. E-mail: yuhangyuan1986@163.com

2011 © The American Computational Science Society. All rights reserved.

mechanics and other fields.

Although Delaunay triangular meshes generation technology now has more mature, but to create large scale and large scale mesh, serial mesh generator in time and memory is difficult to cross the bottleneck. Therefore, from the 1990s, parallel mesh generation has become a new hotspot. R.Lohner proposed by using array face propulsion technology for the parallel mesh generation [1], but after the algorithm in this paper being in domain decomposition, caused the number of the son of the adjacent area is more, searching the boundary is very troublesome. Peraire and Okusanya put forward the data decomposition dichotomy by the modified coordinates, and the number of the mesh is equal to the son of processor [2]. But the process needs to be a large number of units to move between the processor, it still needs some synchronization methods ensure the consistency of the data. Chrisochoides team developed Okusanya and Peraire algorithm [3], and first put forward of the concept of parallel B-W kernel .At the same time of Chrisochoides algorithm in parallel mesh generation, through the exchange processor unit near the border, as the goal of balancing the processor unit quality, dynamically redistributing each processor load, synchronous completed mesh generation and the task of mesh. The algorithm neither avoid the mesh node number of shared pairs treatment nor avoid the need for frequent between processor of communication, serious lowing solution efficiency [4]

As some problems of traditional parallel mesh generation method, this paper puts forward a 2d complex regional parallel mesh generation algorithm, it is based on relations of sub-domain adjacency defining sub-domain, combining sub-domain estimate and dynamic load Figure dividing. And at the same time, through introducing the Dense Circle method, before distributing the dynamic of sub-domain figure to the processor, can ensure these areas not the intersection, that it can effectively reduce the mesh division caused by the heavy price performance, or even eliminate the process. At last, through Delaunay serial mesh generator, and in OpenMP carry numerical experiments, showing the correct result of this parallel Delaunay triangle mesh generation algorithm.

This paper is organized as follows. In section2, the technology of OpenMP is given. An improved parallel Delaunay mesh generation algorithm by using geometric domain decomposition strategy is proposed in section3. The experimental results of the algorithm proposed are given in this paper in section 4.Finally; our work of this paper is summarized in the last section.

1. The technology of OpenMP

OpenMP is an application programming interface (API) designing for writing parallel processing procedure in more on of sharing of storage, by a small compiler command set compositing, including a compilation guidance statements and a used to support its function library [5]. With rapid development of simple and universal, features. OpenMP is portable multithreaded application development of the industry standard, in the fine grain (cycle level) and coarse granularity (function level) thread has high technical efficiency. For the serial application converted into parallel applications, OpenMP instruction is a kind of easy to use and effect powerful tool, it has to make the application for in symmetric processor or more nuclear systems side-by-side execution and get a performance improved potential. Currently Intel C++compiler 10.1, Visual C++ 9.0 and Microsoft Visual Studio 2010 are supported OpenMP. This thesis design process is introducing the use of Microsoft Visual Studio
2010 OpenMP.

2.1 OpenMP parallel implementation model

2011 © The American Computational Science Society. All rights reserved.

OpenMP is a compiler directive and library collection of functions, the compilation directives and library function mainly used for creating shared storage computer parallel programs. At present the latest version of OpenMP standard is 3.0, it supports FORTRAN, C and C++ language, etc. OpenMP in concurrent execution procedures, using "Fork/Join" way [6], the concurrent execution schematic diagram shown as shown in Figure 1

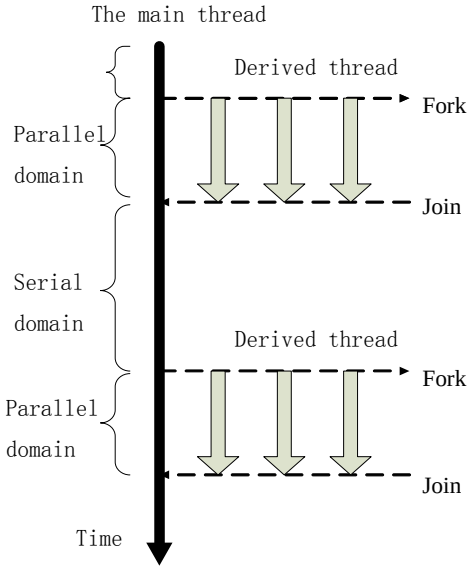

Figure 1 Fork/Join parallel implementation

OpenMP parallel implementation model of the basic idea is: to run the program started to create a main thread (Master), the procedure in the serial part main thread by executive, partly through a parallel derived other threads to execute; But if parallel part not over, is not able to carry out the serial parts. As shown in Figure 1 can see, OpenMP parallel the execution of the program after all is able to implement the parallel part of the behind of the code.

2.2 OpenMP programming model

OpenMP application program interface is a programming model in the structure of shared storage system; it contains three parts, compilation guidance statements (Compiler Directive), operation Library function (Runtime Library) and Environment Variables (Environment Variables). Compiled guidance statements make of instructions and clauses of the list.

Its use format for: # pragma omp parallel [clause[[and]clause]...]New-line structured-block which # pragma omp is compiled the prefix OpenMP legal guidance keywords of key words including parallel, section, sections, task, master, atomic, threadprivate, etc. Clause mainly includes variable share and copy of the concurrent execution. Among these OpenMP legal clauses have default, Shared, copyin, reduction, private, firstprivate, lastprivate, etc. The operation of the OpenMP library function API mainly includes the execution environment function, operation function and operation time lock function three aspects of functions. There are four main OpenMP environment variables [7]: omp_dynamic, omp_threads, omp_nested, and omp_schedule. In parallelizing of serial algorithm , there are many ways, Figure 3 is the implementation of the serial two function test (), use for double circulation executive and join the parallel guidance statements # pragma omp parallel for code executing in parallel.

```
void test()
{
    int a=0;
    for (int i=0;i<100000000;i++)
    {
        a=i+1;
    }
}

int main(int argc, _TCHAR* argv[])
{
    clock_t t1=clock(); //start time
    test();       //serial implementation
    test();
    clock_t t2=clock(); //end time
    printf("Total time=%d\n",t2-t1);

    return 0;
}
```

Figure 2 Serial implementation

2011 © The American Computational Science Society. All rights reserved.

```
void test()
{
    int a=0;
    for (int i=0;i<100000000;i++)
    {
        a=i+1;
    }
}

int main(int argc, _TCHAR* argv[])
{
   clock_t t1=clock();  ///start time
#pragma omp parallel for
//parallel implementation
 for（int j=0;j<2;j++）
   {
    test();
   }
    clock_t t2=clock();  //end time
    printf("Total time=%d\n",t2-t1);

    return 0;
}
```

Figure 3 Parallel implementation

3. Description and analysis of algorithms

3.1 Framework of Delaunay mesh generation

Problem in parallel is the main mode of the parallel mesh generation algorithm, this article is used in parallel, decomposition problem domain for multiple sub-domains, and each sub-domain is mapped to an effective processor, called on serialized mesh algorithm generated subnet mesh.

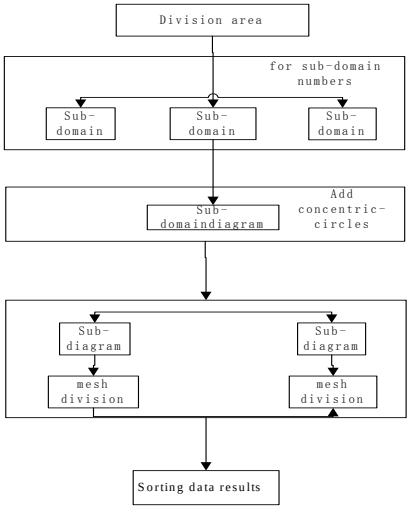

Figure 4 Framework of Delaunay mesh generation

3.2 Algorithm of 2d parallel Delaunay mesh generation

Algorithm of 2d parallel Delaunay mesh generation of the specific steps shown below:

1) for the division of region division, the so-called domain decomposition, will division within the area boundary is circular area for numbers, for example, there are six inner boundary, the division will be divided into seven area, recorded for Ⅰ, Ⅱ, Ⅲ, Ⅳ, Ⅴ, Ⅵ, Ⅶ. Among them the area is according to the distance between the border within and outside the boundaries of distance adjusting.

2) in each of the inner boundary around has previously decorated 10 different radius of the Dense Circles, and then judge the 10 Dense Circle have inters elect with outer boundary or other boundary of e Dense Circle around, the biggest not at the intersection round is the size of the area. If appear at the intersection of situation, it will be the biggest Dense Circle take out, choose its internal Dense Circles in the inner boundary. The inner boundaries can never appear the intersection of situation.

3) Each sub-domain generates mesh independently. The mesh of son domain generation realize in the parallel environment. Let each one processor to deal with an independent sub-domain [8]. The steps we use OpenMP omp_set_num_threads statements by the concurrent execution set code thread number, make a separate thread to generate Delaunay triangle mesh. Serialization Delaunay mesh generation algorithm, when parallel mesh generate, need a single son region for serial mesh generation, this paper adopted based on Bowyer-Watson with the insertion point of the realization of the kernel a 2d serialization Delaunay mesh generator, it can generate mesh in a faster pace in the single mesh.

4) Processor communications complete the border area outside of the mesh partition.

5) Sorting data results.

2011 © The American Computational Science Society. All rights reserved.

4. Experimental result and analysis

4.1 Result of experiment

In order to verify the correctness and effectiveness of the algorithm, selected round and rectangular, trapezoid and irregular graphics as outer boundary, the inner boundary is a circle, and to have an experiment with the improvement of the parallel Delaunay triangular mesh algorithm, the experimental results as shown in Figure 5, Figure 6, Figure 7, Figure 8 Our algorithm is implemented in C++. Hardware Environment mainly consists of Inter(R)Core(TM)i3CPU、2.53GHz、Memory2G、Video Memory512M,and Operating System is Windows 7 Ultimate.

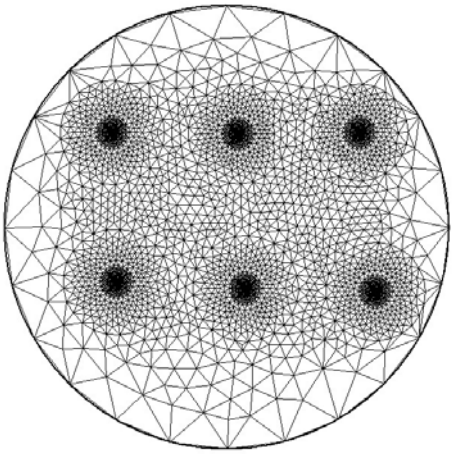

Figure 5 Outer boundary is circular area; the inner boundary is a circle

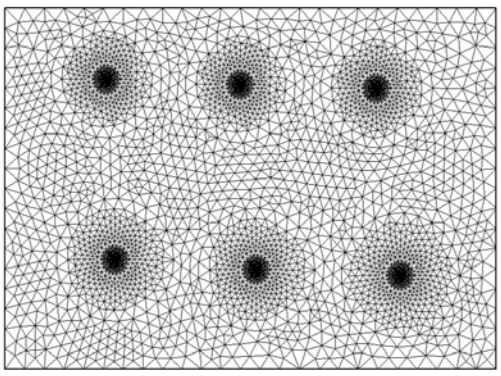

Figure 6 Outer boundary is the rectangular area; the inner boundary is a circle

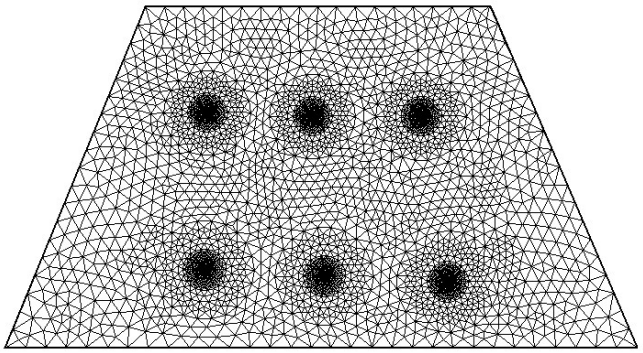

Figure 7 Outer boundary is trapezoid area; the inner boundary is a circle

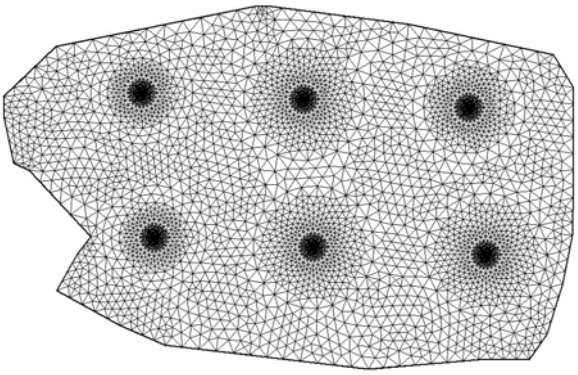

Figure 8 Outer boundary is any irregular area; the inner boundary is a circle

4.2 Delaunay algorithm parallelization and performance analysis

As the program execution time uncertainty, each time running time not be sure, as the example of the Figure 7 for five times operation to the programs gain the serial execution time and parallel time. According to speed-up ratio formula $S = T_{seq} / T_p$, which S speed-up, Tseq for serial execution time,

2011 © The American Computational Science Society. All rights reserved.

Tp for concurrent execution parallel time, that speed-up ratio data (see table 1) to analyze the parallel performance.

Table 1 Running schedule.

n	Tseq/ (10^{-3}s)	Tp/s (10^{-3}s)	S
1	2104	1738	1.216
2	2165	1717	1.260
3	2128	1727	1.232
4	2133	1729	1.233
5	2188	1769	1.236
Ave	2143	1736	1.234

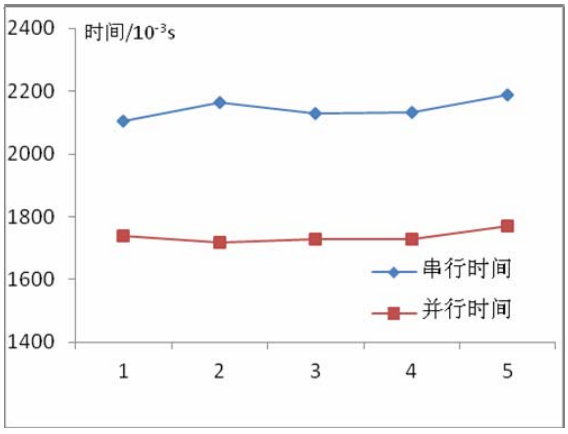

Figure 9 Time contrast

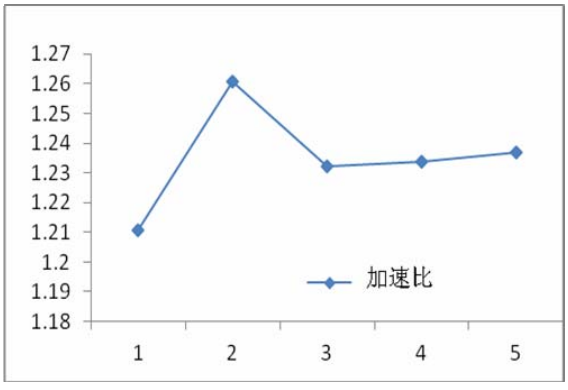

Figure 10 Speed-up ratio distributions

Analyzing the existing data, in order to have certain reference data is analyzed, the serial and parallel comparisons is shown in Figure 9 and Figure 10 speedup ratio data. From the data of the Figure 9 and Figure 10, we can get the following conclusion, due to the instability of the computer, no matter in serial running time or in parallel running time, each time the results are not the same, in the code Delaunay mesh generation parallelism, took five data, through the analysis, we can see that running time had certain reduce, realized the purpose of mesh generation. For a complicated Delaunay mesh generation program, parallelization its part of the codes, on the dual-core processor computer, accelerate than achieve 1.234, the performance has been some improvement.

5. Conclusions

In view of the traditional parallel mesh generation method shortcomings; this paper presents an improved two-dimensional parallel Delaunay mesh generation algorithm. By introducing the method of Dense Circle, before a subdomain of dynamically assigned to the processor, to ensure that the area does not exist this situation, thereby effectively reducing re-meshing performance caused by price, or even completely eliminate the process. Experimental results showing that the parallel Delaunay triangular mesh generation algorithm is correct. The next step, can consider to deal with more complex two-dimensional complex regional mesh, can also be considered based on the current algorithm, study under the condition of three-dimensional region parallel Delaunay triangle mesh generation algorithm.

Acknowledgments

The authors are grateful for the financial support provided Important National Science & Technology Specific Projects 2011ZX05038-003 and Beijing Natural Science Foundation 4112016 and Funding

2011 © The American Computational Science Society. All rights reserved.

Project for Academic Human Resources Development in Institutions of Higher Learning under the Jurisdiction of Beijing Municipality PHR201008239.

References and Notes

[1] Lohner.Rainald. A parallel advancing front grid generation scheme [J].Int J Num Meth Eng, 2001, 51:663-678.

[2] T.OKusanya and J.Peraire.Parallel Unstructured Mesh Generation. Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Fluid Dynamic and Related Fields, Mississippi State Universtiy, MS, USA, 1996.

[3] Chrisochoides N. Parallel Mesh Generation. In Bruaset AM Tveito A eds. Num erical Solution of Partial Differential Equations on Parallel Computers. Spring, 2006.pp, 237-266.

[4] J.J. Chen, Unstructured Mesh Generation and its Parallelization [D]. The doctoral dissertation of Zhejiang University, 2006.

[5] http://www.openmp.org

[6] W.M. Zhou, Multi-core computing and programming [M]. Huazhong University of Science and Technology press 2009.

[7] Y. Dong, Improved Energy-Optimal OpenMP Static Scheduling Algorithm [J]. Journal of Software, 2011, 22(9):2235-2247.

[8] X. Liu, Research on Pre-processing Methods of Unstructured Grids [J].Computer Science, 2012, 39(3):308-311.

2011 © The American Computational Science Society. All rights reserved.