



A multigrain Delaunay mesh generation method for multicore SMT-based architectures

Christos D. Antonopoulos^a, Filip Blagojevic^d, Andrey N. Chernikov^{c,*}, Nikos P. Chrisochoides^c, Dimitrios S. Nikolopoulos^b

^a Department of Computer and Communications Engineering, University of Thessaly, Volos, Greece

^b Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, United States

^c Department of Computer Science, The College of William and Mary, Williamsburg, VA 23187, United States

^d Lawrence Berkeley National Lab, Berkeley, CA 94720, United States

ARTICLE INFO

Article history:

Received 31 August 2006

Received in revised form

31 December 2008

Accepted 24 March 2009

Available online 14 April 2009

Keywords:

Parallel
Mesh Generation
Delaunay
Multigrain
Multicore
SMT

ABSTRACT

Given the proliferation of layered, multicore- and SMT-based architectures, it is imperative to deploy and evaluate important, multi-level, scientific computing codes, such as meshing algorithms, on these systems. We focus on Parallel Constrained Delaunay Mesh (PCDM) generation. We exploit coarse-grain parallelism at the subdomain level, medium-grain at the cavity level and fine-grain at the element level. This multi-grain data parallel approach targets clusters built from commercially available SMTs and multicore processors. The exploitation of the coarser degree of granularity facilitates scalability both in terms of execution time and problem size on loosely-coupled clusters. The exploitation of medium-grain parallelism allows performance improvement at the single node level. Our experimental evaluation shows that the first generation of SMT cores is not capable of taking advantage of fine-grain parallelism in PCDM. Many of our experimental findings with PCDM extend to other adaptive and irregular multigrain parallel algorithms as well.

Published by Elsevier Inc.

1. Introduction

As modern supercomputers integrate more and more processors into a single system, system architects tend to favor layered multiprocessors, since such designs seem to be at the sweet-spot of the cost/performance tradeoff. Most machines in the Top500 list [54] are clusters, often consisting of small-scale SMP nodes. The recent commercial success of simultaneous multithreaded (SMT) processors [41,56] and multicore processors (CMP) [36] with scalar, superscalar, or SMT cores [33], introduces additional levels in parallel architectures, since more than one thread can co-execute on the same physical processor, sharing some or all of its resources. The efficient exploitation of the functionality offered by these layered architectures introduces new challenges for

application developers. Applications that expose multiple levels of parallelism, at different granularities, appear as ideal candidates for the exploitation of the opportunities offered by layered multiprocessors. However, developers have to target both micro-scalability across the multiple execution contexts of each physical processor, and macro-scalability across processors or different nodes of the system.

This paper focuses on the design and implementation of parallel mesh generation algorithms and software on such multilevel architectures. Parallel mesh generation is essential in many scientific computing applications in health care, engineering, and science. Such applications often require the creation of meshes with size in the order of billions of elements [34]. Our study provides a macroscopic understanding of the behavior of mesh generation codes on modern parallel architectures. It is a step towards meeting the time and quality constraints set by real-world applications [18]. Moreover, the results of our study are valid in the context of not only finite element mesh generation methods, but also in the context of other asynchronous multigrain, parallel algorithms.

Parallel mesh generation procedures decompose the original mesh generation problem into smaller subproblems that can be solved (meshed) in parallel. The subproblems can be formulated to be either tightly or partially coupled or even decoupled. The

* Corresponding author.

E-mail addresses: cda@inf.uth.gr (C.D. Antonopoulos), fblagojevic@lbl.gov, filip@cs.vt.edu (F. Blagojevic), ancher@cs.wm.edu (A.N. Chernikov), nikos@cs.wm.edu (N.P. Chrisochoides), dsn@cs.vt.edu (D.S. Nikolopoulos).

URLs: <http://inf-server.inf.uth.gr/~cda> (C.D. Antonopoulos), <http://www.cs.vt.edu/~filip> (F. Blagojevic), <http://www.cs.wm.edu/~ancher> (A.N. Chernikov), <http://www.cs.wm.edu/~nikos> (N.P. Chrisochoides), <http://www.cs.vt.edu/~dsn> (D.S. Nikolopoulos).

Fig. 1. (a) A coarse-grain decomposition into 128 subdomains of a cross-section of a regenerative cooled pipe model. (b) Medium-grain parallel expansion of multiple cavities within a single subdomain. (c) Fine-grain parallel expansion of a single cavity ($C \cdot p_i / l$) by concurrent testing of multiple triangles ($\Delta \cdot p_k p_l p_m / l$, $\Delta \cdot p_m p_n p_s / l$, $\Delta \cdot p_s p_t p_k / l$).

coupling of the subproblems (i.e., the degree of dependency) determines the intensity of the communication and synchronization between processing elements working on different subproblems. The three most widely used techniques for parallel mesh generation are Delaunay, Advancing Front, and Edge Subdivision [25]. In this paper, we use the Delaunay technique because it can mathematically guarantee the quality of the mesh. More specifically, we focus on Constrained Delaunay meshing [15]. The sequential execution time of our parallel implementation (PCDM) is comparable to that of the best (to our knowledge) sequential implementation [51]. At the same time, PCDM explores concurrency at three levels of granularity: (i) *coarse-grain* at the subdomain level, (ii) *medium-grain* at the cavity level, and (iii) *fine-grain* at the element level. Fig. 1 depicts all three levels of granularity, one for each level of parallelization.

In the coarse-grain parallel implementation, the domain is decomposed [39] into $N \gg P$ subdomains, where P is the number of processors (Fig. 1a). $\frac{N}{P}$ subdomains are mapped, using METIS [35], to processors in way that the ratio of interfaces to area is minimized i.e., improve affinity by assigning neighbor subdomains to a single thread, core or processor. In the medium-grain parallel implementation, multiple cavities are expanded concurrently by multiple threads. Each thread expands the cavity of a bad-quality triangle. As soon as each cavity has been calculated, its triangles are deleted and the cavity is retriangulated. In order to preserve the conformity of the mesh, the algorithm has to ensure that there are no conflicts between concurrently expanded cavities. In other words, concurrently expanded cavities are not allowed to share triangles. Finally, in the fine-grain parallel implementation, multiple threads work together for a single cavity expansion and thus the degree of parallelism is limited to three for 2-dimensions, and four for 3-dimensions.

We investigate parallelization approaches for mapping algorithmic multi-grain concurrency to different parallel execution layers in hardware, for clusters built from: (1) conventional, single-thread, single-core processors, and (2) clusters built from SMP nodes with commercially available layered CMP/SMT processors. Our experimental evaluation shows that the coarse-grain, MPI-based approach proves scalable across large numbers of loosely coupled cluster nodes, across different processors within each node, and – under certain configurations – even across cores within each processor. The coarse granularity of PCDM is an effective means of either reducing execution time, for time-sensitive applications, or of increasing the maximum problem size that can be tackled, by facilitating the exploitation of the total memory and processing power available on multiple nodes. The medium-grain of concurrency in PCDM offers a high degree of parallelism, at an exploitable granularity by today's SMT processors. It allows the effective use of SMT contexts and results in a reduction of execution time on a single core. Finally, we find that the fine-grain parallelism in PCDM is not exploitable on top of the SMTs we use in this study,

due to synchronization overhead and lack of hardware support for light-weight thread management.

This paper identifies conditions under which a multilevel, multigrain, parallel mesh generation code can effectively exploit the performance potential of current and emerging multithreaded architectures. Our study also raises the level of understanding for the limitations present when developing efficient parallel algorithms and software for asynchronous, adaptive and irregular applications on current and emerging multilevel parallel architectures.

The rest of the paper is organized as follows. Section 2 discusses previous work. In Section 3 we describe the sequential Delaunay meshing algorithm. Section 4 discusses the parallel multi-level and multi-granular PCDM algorithm. In Section 5 we present the implementation and evaluation of the coarse-, medium-, and fine-grain approaches on real systems. Finally, Section 6 summarizes the paper.

2. Related work

In this section, we provide a brief coverage of related work on parallel mesh generation and on innovative parallel architectures that enable single-chip multithreaded execution, either across multiple cores, or within a single core.

2.1. Parallel mesh generation

In [20] we presented an exhaustive review of parallel mesh generation methods. In this section we focus only on parallel methods which are directly related to PCDM.

The coarse-grain parallel mesh generation method we study in this paper is weakly coupled at its outer level of parallelism, i.e., asynchronous with small messages, and exhibits low communication costs. This method is among the four different parallel mesh generation classes of methods that were developed in our group: (1) tightly coupled [19,45], (2) partially coupled [11–13], (3) weakly coupled [14,17], and (4) decoupled [38,39].

In [11,13] we presented a theoretical framework and the experimental evaluation of a partially coupled parallel Delaunay refinement (PDR) algorithm for the construction of the uniform guaranteed quality Delaunay meshes. We then extended the PDR approach [10,12] for the non-uniform case, when the element size is controlled by a user-defined function. The non-uniform PDR algorithm has been implemented on shared memory and offers the possibility to vary the granularity of the refinement units. However, its experimental evaluation is still in progress. In the current study, we focus on the weakly coupled PCDM method [14] with the goal of exploiting multiple levels of parallelism and mapping them efficiently on emerging multilevel parallel architectures.

In [38,39] we presented the Parallel Delaunay Domain Decoupling (PD^3) method. PD^3 completely decouples the individual

