

Research paper

Parallelized implementation of an explicit finite element method in many integrated core (MIC) architecture

Cai Yong, Li Guangyao*, Liu Wenyang

State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082, China
 Joint Centre for Intelligent New Energy Vehicle, Hunan University, Changsha 410082, China

ARTICLE INFO

Keywords:

Explicit finite element
 MIC
 Intel Xeon Phi coprocessor
 Parallel computing
 Nonlinear analysis

ABSTRACT

Hardware accelerators are becoming increasingly important in boosting high performance computing systems. In this study, we develop a parallel explicit finite element (FE) analysis system based on a many integrated core (MIC) architecture for fast simulation of nonlinear dynamic problems of plate and shell structures. To minimize data transfer between heterogeneous architectures, parallel computation of the all explicit FE calculation is realized by developing a vectorized thread-level parallelism algorithm. The parallelism includes a novel dependency relationship link based method for efficiently solving parallel explicit shell element equations. A heterogeneous model is established to overlap data transfer and offloaded computation, and thus reduce the time required for large intermediate data storage in the actual engineering nonlinear problem simulation. Finally, a high performance nonlinear dynamic simulation system is developed. The simulations of benchmarks and engineering problems show that the parallel computing method proposed in this paper can give full play to the hardware performance of MIC architecture and effectively improve the computation efficiency of an explicit FE solution. For a bus body model containing approximately 3.8 million degrees of freedom, the computational speed is improved 17 times over CPU sequential computation, and the relative speedup grows with the increasing number of threads, the highest relative speedup exceeds 80.

1. Introduction

With the maturity of various numerical techniques, finite element (FE) based CAE analysis technology has a wider range of applications in industrial product design because it can effectively shorten the cycle of new product development. For example, explicit shell element based CAE software plays an important role in automotive design [1]. To improve computational efficiency, various parallel computing methods have been proposed, improved and applied to practical engineering and scientific problems [2]. With the development of computer technology, there has been a substantial increase in the use of heterogeneous systems for parallel computing because of their high level of computational integration [3]. In a heterogeneous computing system, the compute-intensive and parallelizable tasks can be offloaded through the Peripheral Component Interconnect Express (PCI-E) to coprocessors, such as the Graphics Processing Units (GPUs) and the Many Integrated Core (MIC) coprocessors, for efficient parallel execution [4]. Heterogeneous computing systems based on these accelerators have been adopted by most mainstream high-performance computers, for example, the No. 1 Supercomputer Milky Way-2 developed by China

contains 48,000 Intel Xeon Phi 31S1P MIC coprocessors [5].

Compared to the MIC coprocessor, the GPU-based parallel applications have entered the mature period. Since GPUs with high-level programming language support were first introduced by Nvidia in 2006, several groups have applied them to FE analysis and nonlinear dynamic simulations, and achieved good speedup ratios [6–9]. A general-purpose graphical processing unit (GPGPU) is employed in most modern accelerators based on a large number of streaming multi-processors and expressed using a specialized programming model to achieve high theoretical peak performance. Therefore, difficulties still exist in designing high-efficient and high-precision GPU-based parallel computing programs, especially for applications such as nonlinear FE analysis of large scale plate and shell structures that contain complex mathematical instructions and require a sophisticated design of memory read and write [10].

To address these issues, new parallel explicit FE code in heterogeneous environments with a multi-cores CPU and an Intel Xeon Phi coprocessor is developed in this paper. This code is designed to accelerate the simulation of nonlinear dynamic problems of plate and shell structures, especially for automotive body design. The MIC architecture

* Corresponding author at: Hunan University, State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Changsha 410082, China.
 E-mail addresses: caiyong@hnu.edu.cn (Y. Cai), gyli@hnu.edu.cn (G. Li).

uses x86-compatible cores with high performance using standard, parallel programming models and languages [11]. Therefore, MIC parallel computing power has gradually attracted the attention of researchers in the field of CAE analysis. Rao [12] presented parallel computational strategies to implement explicit nonlinear finite element analysis code onto distributed memory parallel computers for solving large-scale problems in structural dynamics. Kružel and Banaš [13] and Kružel and Banaš [14] presented an implementation of the FE numerical integration algorithm for the Xeon Phi coprocessor, where an OpenCL based parallel algorithm is developed. Banaš et al. [15] presented investigations on the implementation and performance of the finite element numerical integration algorithm for first order approximations and three processor architectures, popular in scientific computing, classical CPU, Intel Xeon Phi and NVIDIA Kepler GPU. Tak and Park [16] investigated a domain decomposition method of the FE method using MIC architecture in order to determine the most effective MIC usage. Saule et al. [17] investigated the performance of the Xeon Phi coprocessor for various sparse linear algebra kernels. In addition, there are some mature applications in other areas. For example, Weinberg et al. [18] ported an MPI-based real-world geophysical application to the MIC architecture; Kang et al. [19] proposed a parallel photon searching algorithm by using a radiance estimation approach for coherent shading points on the MIC Architecture. These researches are effective in improving the efficiency of computing in their respective fields. Unfortunately, it is still rare to apply the MIC architecture application to the nonlinear dynamic problem of plate and shell structure, although this is a typical time-consuming problem.

The remainder of this paper is organized as follows. In Section 2, the MIC architecture and the Intel Xeon Phi 5110P coprocessor are briefly introduced, and a simple performance test is presented. In Section 3, the explicit FE method and its sequential implementation are introduced. In Section 4, the parallel implementation of our simulation system based on MIC architecture is introduced in detail. Numerical experiments are used to evaluate the performance of our parallel simulation system in Section 5. Finally, concluding remarks are presented in Section 6.

2. Overview of the MIC architecture and its performance test

The basis of the Intel MIC architecture is to create an x86-compatible multiprocessor architecture that can use existing parallelization software tools. Programming tools include OpenMP, OpenCL and specialised versions of Intel's Fortran, C++ and math libraries [20]. In this work, two Xeon Phi products based on 22 nm Knights Corner cores are used as the MIC coprocessors, including Xeon Phi 5110P and Xeon Phi 3120A, both of which are PCI-e cards outfitted with a single coprocessor and several gigabytes of GDDR5 memory. And the common hardware resources include x86 ISA, 4-way SMT per core, 512-bit SIMD units, 32KB L1 instruction cache, 32KB L1 data cache, coherent L2 cache with 512KB per core, and ultra-wide ring bus connecting processors and memory. Most of the performance of the MIC architecture comes from the vector processing unit (VPU). The VPU can perform many basic instructions, such as addition or division, and mathematical operations, such as $\sin()$ and $\sqrt{}$, allowing 8 double precision operations per cycle. The VPU can also perform both an addition and a multiplication simultaneously using a Fused Multiply Add (FMA) instruction.

To determine whether MIC architecture is suitable for the parallel computing of explicit FE, this paper designs a simple two-dimensional bar problem as shown in Fig. 1. One end of the bar is fixed and the other

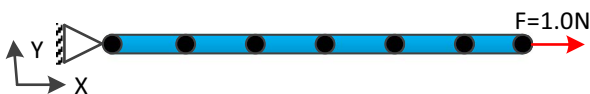


Fig. 1. A simple bar problem.

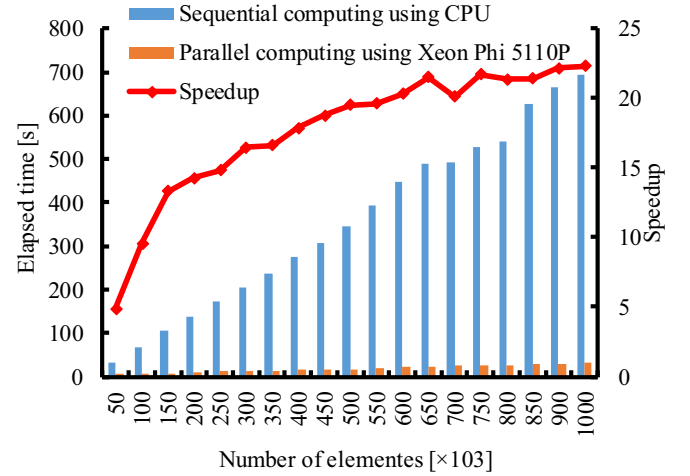


Fig. 2. Comparison of elapsed time and speedup using Xeon Phi 5110P.

end is subjected to an axial force of 1.0 N. Although the problem is very simple, it contains a complete explicit FE calculation process. Compiler assisted offload is used to achieve simple parallel computing. The elapsed time of 10,000 iterations and speedup obtained by parallel calculation using Xeon Phi 5110P is shown in Fig. 2. As seen, such a simple case can achieve a significant improvement in computational efficiency. The maximum speedup is 23. Therefore, it is envisioned that the migration of complex explicit FE analysis of plate and shell structures to the MIC architecture will also achieve good computational efficiency.

3. Explicit FE method and its sequential implementation

3.1. Central difference algorithm

The most commonly used explicit time integration technique is the central difference algorithm. Consider the global equation of FE given by:

$$\mathbf{M}\ddot{\mathbf{u}}_t = \mathbf{F}_{ext} - \mathbf{F}_{int} \quad (1)$$

where \mathbf{M} is the mass matrix, \mathbf{u} is the node displacement vector, \mathbf{F}_{int} is the nodal internal force, \mathbf{F}_{ext} is the external force. The solution for the next time step $t + \Delta t$ can be obtained according to the central difference formula by employing Eq. (1) for the known configuration at time t :

$$\dot{\mathbf{u}}_t = \frac{1}{2\Delta t}(\mathbf{u}_{t+\Delta t} - \mathbf{u}_{t-\Delta t}) \quad (2)$$

$$\ddot{\mathbf{u}}_t = \frac{1}{\Delta t^2}(\mathbf{u}_{t+\Delta t} - 2\mathbf{u}_t + \mathbf{u}_{t-\Delta t}). \quad (3)$$

Substitute Eqs. (2) and (3) into Eq. (1)

$$\frac{1}{\Delta t^2}\mathbf{M}\mathbf{u}_{t+\Delta t} = \mathbf{F}_{ext} - \mathbf{F}_{int} + \mathbf{M}\frac{1}{\Delta t^2}(2\mathbf{u}_t - \mathbf{u}_{t-\Delta t}) \quad (4)$$

when $\mathbf{u}_{t-\Delta t}$ and \mathbf{u}_t are given, the displacement $\mathbf{u}_{t+\Delta t}$ can be obtained. Therefore, Eq. (4) is the recursive formula for solving at each discrete-time point, where \mathbf{M} is a diagonal matrix. The calculation process avoids the solution of the equations, which is the most important advantage of the explicit algorithm. However, the convergence of the explicit algorithm is restricted by the conditional stability. For a damped equation of motion, the critical time step is given in terms of the highest eigenvalue in the system [21,22]:

$$\Delta t \leq \frac{2}{\omega_{max}}(\sqrt{1 + \varepsilon^2} - \varepsilon) \quad (5)$$

where ω_{max} is the maximum frequency of the system, ε is the fraction of critical damping in the highest mode staying within the stability limit.

Download English Version:

<https://daneshyari.com/en/article/6961397>

Download Persian Version:

<https://daneshyari.com/article/6961397>

[Daneshyari.com](https://daneshyari.com)