Advances in Engineering Software 85 (2015) 108–124

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

A multiple-GPU based parallel independent coefficient reanalysis method and applications for vehicle design

Guanqiang He [a], Hu Wang [a,*], Enying Li [b], Guanxin Huang [a], Guangyao Li [a,*]

[a] State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha 410082, PR China
[b] School of Logistics, Central South University of Forestry and Teleology, Changsha 41004, PR China

ARTICLE INFO

Article history:
Received 25 November 2014
Received in revised form 27 January 2015
Accepted 10 March 2015
Available online 29 March 2015

Keywords:
Parallel
GPU
MPI
Reanalysis
Vehicle design

ABSTRACT

The main limits of reanalysis method using CUDA (Compute Unified Device Architecture) for large-scale engineering optimization problems are low efficiency on single GPU and memory bottleneck of GPU. To breakthrough these bottlenecks, an efficient parallel independent coefficient (IC) reanalysis method is developed based on multiple GPUs platform. The IC reanalysis procedure is reconstructed to accommodate the use of multiple GPUs. The matrices and vectors are successfully partitioned and prepared for each GPU to achieve good load balance as well as little communication between GPUs. This study also proposes an effective technique to overlap the computation and communication by using non-blocking communication strategy. GPUs would continue their succeeding tasks while communication is still carried out simultaneously. Furthermore, the CSR format is used in each GPU for saving the memory. Finally, large-scale vehicle design problems are implemented by the developed solver. According to the test results, the multi-GPU based IC reanalysis method has potential capability for handling the real large scale problem and reducing the design cycle.

1. Introduction

One of the main obstacles in the structural optimization is high computational cost due to the repetitive computations during optimization procedure. Therefore, reanalysis methods have been motivated by the concept that the results of a modified structure can be approximated by an initial complete analysis instead of solving the complete set of modified equations. The most popular reanalysis method might be the combined approximations (CA) method proposed by Kirsch [1]. At the early stage, it was used only for linear reanalysis [2,3]. Sequentially, Leu and Huang [4] attained accurate results and significant reduction in computation effort when they applied CA method for complex nonlinear analysis problems. Furthermore, reanalysis for eigenvalue problems based on the CA method has also been developed in the early 2000s [5,6]. However, most of cases solved in these literatures are space truss structures or other simple problems, and the sizes of these cases are limited. In our previous study, the CA method is difficult to address large-scale problems due to its low efficiency of matrix decomposition operations, so a novel reanalysis method named the independent coefficients (IC) method was proposed by Huang [7] for handling large-scale real engineering problems. Since some operations such as stiffness matrix decomposition are not involved in this method, computational cost of IC is significantly improved. However, it should be pointed out that the computation demanding will rise rapidly if the scale of reduced equilibrium system increases. Therefore, the computational cost is still expensive especially for large-scale problems. The computational cost is mainly caused by the matrix vector multiplication, a dominative operation in the IC method. Other operations (such as vector dot product and scalar operation) are also sensitive to the efficiency of reanalysis. In order to solve large scale engineering optimization problems, the efficiency of IC method would desire to be improved further. Obviously, the most direct way of improving the efficiency is parallelism.

Parallel computing can accelerate science applications tremendously and has become a dynamic field with continuous innovation in computing hardware and programming model. Numerous parallel applications have been implemented on supercomputers [8], but these platforms are extremely expensive. One emerging technology for parallel computing is to port the parallel part of the calculations to a graphics processor unit (GPU) [9]. Compared with supercomputers including numerous CPUs, the use of GPU makes it is possible to get considerable parallel throughput on a much cheaper desktop computer rather than relying on large computer clusters. Currently multi-threaded GPUs can achieve several

* Corresponding author. Tel.: +86 731 88821445; fax: +86 731 88821807.
E-mail addresses: wanghu@hnu.edu.cn (H. Wang), gyli@hnu.edu.cn (G. Li).

http://dx.doi.org/10.1016/j.advengsoft.2015.03.006
0965-9978/© 2015 Elsevier Ltd. All rights reserved.

Teraflops of peak computing power that are orders of magnitude greater than that of modern CPUs by means of a combination of hundreds of parallel processors on graphics card, and 20 times or more higher memory bandwidth than CPU to memory interface in commodity computers. The many-core GPU with these hardware advantages enables programmers to achieve speedups of an order of magnitude over a standard CPU in various domains [10–12] and is growing in popularity [13,14]. Several GPU programming interfaces such as NVIDIA's compute unified device architecture (CUDA) [15], and open computing language (OpenCL) [16] have been developed to fascinate the programmers while exposing enough of the hardware to allow the resources to be well utilized. As a high-level programming model, CUDA provides a simplified C-based programming language in which most of C language grammars are applicable, and it has been used in various scientific computing domain [17,18]. In order to improve the efficiency of reanalysis problems, Wang firstly presented a parallel CA method by using single GPU to improve the computational efficiency [19].

Although the use of single GPU system makes it possible to improve the performance of several reanalysis problems, many significantly large-scale engineering problems suggest a combination of the power of multiple GPUs to overcome memory limitations of single GPU and to achieve higher efficiency. As a standard library definition for message passing in parallel computer system, message passing interface (MPI) [20] is investigated and employed to establish communications between GPUs. Each CPU core runs a separate process and controls one of the available GPUs. The memory limitations of a single GPU can be overcome by suitable distributing the data among the GPUs, and it is flexible to scale the parallel computing model according to the number of nodes which is equipped with a single GPU or multiple GPUs. Since direct methods used in reanalysis method involves operations like matrix decomposition, and these are not suitable to be implemented on multiple GPUs for memory access pattern. In this study, the iterative method, preconditioned conjugate gradient (PCG) method is chose as the solver for linear system in IC algorithm. It should be noted that many matrix and vector multiplication operations are involved in the PCG solver, and these operations are perfectly suitable to run on multiple GPUs for their fine parallelism granularity.

In this study, a MPI-CUDA implementation of IC method is developed for multiple GPUs. The IC procedure is reconstructed to accommodate the use of multiple GPUs. To improve the scale of solution with IC reanalysis method, this paper efficiently utilizes the CSR format for stiffness matrix storage. The stiffness matrix and corresponding vectors are successfully partitioned on multiple GPUs to achieve good task balance as well as few data transfer, and the multiple GPUs cooperate with each other well to obtain the final optimization results. The proposed MPI-CUDA implementation also incorporates an efficient programming strategy to successively boost the parallel performance by using non-blocking communication MPI functions to overlap the CPU data transfer and GPU computation.

The rest of this paper is organized as follows. In Section 2, the IC method is briefly reviewed. In Section 3, the parallel IC method based on MPI and CUDA will be presented. Section 4 will give some numerical examples to demonstrate the effectiveness of the proposed method. Finally, some conclusions are given.

2. Independent coefficients method

Given a positive-definite symmetric stiffness matrix $\mathbf{K}_0$ and the load vector $\mathbf{R}_0$ of an initial design of structure, the resulting displacements $\mathbf{r}_0$ can be solved by the equilibrium equations.

$$\mathbf{K}_0\mathbf{r}_0 = \mathbf{R}_0 \tag{1}$$

A change in the design will lead to the modified stiffness matrix $\mathbf{K}$ and the modified load vector $\mathbf{R}$ can be given by

$$\mathbf{K} = \mathbf{K}_0 + \Delta\mathbf{K} \tag{2}$$

$$\mathbf{R} = \mathbf{R}_0 + \Delta\mathbf{R} \tag{3}$$

where $\Delta\boldsymbol{K}$ and $\Delta\boldsymbol{R}$ refer to the changes in the stiffness matrix and the load vector, respectively. The purpose of reanalysis method is to find efficient and high quality expressions of the displacements $\boldsymbol{r}$ of modified structure without solving the complete set of modified analysis equations.

$$\mathbf{Kr} = \mathbf{R} \tag{4}$$

The IC method [7] replaces solving Eq. (4) directly with calculating the displacements of the DOFs influenced by the changes. The main strategy of IC method is briefly summarized as follows.

The modified displacements $\mathbf{r}$ is assumed by the following expression.

$$\mathbf{r} = \mathbf{r}_0 + \Delta\mathbf{r} \tag{5}$$

Then Eq. (4) can be written as

$$\mathbf{K}(\mathbf{r}_0 + \Delta\mathbf{r}) = \mathbf{R} \tag{6}$$

Rewrite Eq. (6) in the following expression.

$$\mathbf{K}\Delta\mathbf{r} = \mathbf{R} - \mathbf{K}\mathbf{r}_0 \tag{7}$$

Define $\boldsymbol{\delta}$ be the residual value of the initial displacements $\boldsymbol{r}_0$ as Eq. (8).

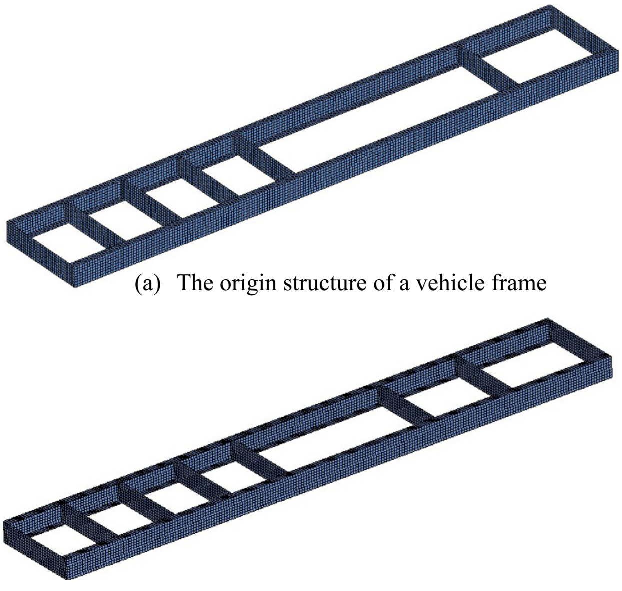
$$\boldsymbol{\delta} = \mathbf{R} - \mathbf{K}\mathbf{r}_0 \tag{8}$$

Eq. (7) becomes

$$\mathbf{K}\Delta\mathbf{r} = \boldsymbol{\delta} \tag{9}$$

Since the structural modification usually is local, only few elements of vector $\boldsymbol{\delta}$ are non-zero. In the coming step of IC method, all the DOFs related to the non-zero elements of $\boldsymbol{\delta}$ would be recorded by pre-selecting a small tolerance ε. In practice, if

$$|\boldsymbol{\delta}(i)| > \varepsilon, \tag{10}$$


(a) The origin structure of a vehicle frame

(b) An local modification for the vehicle frame

Fig. 1. An illustration of IC application.