Computer Physics Communications 193 (2015) 38–48

Contents lists available at ScienceDirect

Computer Physics Communications

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

Parallel finite element technique using Gaussian belief propagation

Yousef El-Kurdi [a,*,1], Maryam Mehri Dehnavi [b,1], Warren J. Gross [a], Dennis Giannacopoulos [a]

[a] McGill University, Department of Electrical and Computer Engineering, 3480 University Street, Montreal, Quebec, H3A 0E9, Canada

[b] Massachusetts Institute of Technology, G770, 32 Vasser Street, Cambridge, MA 02139, USA

ARTICLE INFO

Article history:
Received 29 January 2015
Received in revised form
2 February 2015
Accepted 30 March 2015
Available online 6 April 2015

Keywords:
FEM
Gaussian belief propagation
Graphical models
Gaussian distributions
Parallel algorithms
GPU

ABSTRACT

The computational efficiency of Finite Element Methods (FEMs) on parallel architectures is severely limited by conventional sparse iterative solvers. Conventional solvers are based on a sequence of global algebraic operations that limits their parallel efficiency. Traditionally, sophisticated programming techniques tailored to specific CPU architectures are used to improve the poor performance of sparse algebraic kernels. The introduced FEM Multigrid Gaussian Belief Propagation (FMGaBP) algorithm is a novel technique that eliminates all global algebraic operations and sparse data-structures. The algorithm is based on reformulating the FEM into a distributed variational inference problem on graphical models. We present new formulations for FMGaBP, which enhance its computation and communication complexities. A Helmholtz problem is used to validate the FMGaBP formulation for 2D, 3D and higher FEM degrees. Implementation techniques for multicore architectures that exploit the parallel features of FMGaBP are presented showing speedups compared to open-source libraries, specifically deal.II and Trilinos. FMGaBP is also implemented on manycore architectures in this work; Speedups of 4.8X, 2.3X and 1.5X are achieved on an NVIDIA Tesla C2075 compared to the parallel CPU implementation of FMGaBP on dual-core, quad-core and 12-core CPUs respectively.

1. Introduction

Manycore architectural advances in High Performance Computing (HPC) have introduced difficult challenges to the FEM software design. The conventional FEM software relies on performing global and sparse algebraic operations that severely limits its parallel performance. Many attempts were made to improve the performance of conventional sparse computations at the expense of sophisticated programming techniques. Such techniques are tailored to specific CPU hardware architectures, such as cache access optimizations, data-structures and code transformations [1]. These optimizations are known to limit code portability and reusability. For example, implementations of Conjugate Gradient (CG) solvers for FEM problems [2], require global sparse operations which perform at a low fraction of the peak CPU computational throughput [3]. Also accelerating CG solvers on parallel architectures is communication-bound; recent attempts to improve the communication overhead of such solvers through reformulation, namely communication avoiding schemes, suffer from numerical instability and limited support for preconditioners [4,5]. This performance bottleneck is even more pronounced when high accuracy FEM analysis scales to large number of unknowns, in the order of millions or more, which prevents the FEM software users from productively utilizing their parallel HPC platforms.

While existing generic and optimized libraries such as deal.II [6], GetFEM++ [7], and Trilinos [8] can be used for sparse FEM computations; obtaining a sustained performance can be difficult due to the varying sparsity structure for different application areas. In addition, such libraries do not help with the costly stage of assembling the sparse matrix. However, recent work by Kronbichler et al. [9] use a matrix free (MF) approach to execute the sparse matrix–vector multiply (SMVM) kernel in the CG solver. While their approach shows promising speedups, it does not depart from the sequential global algebraic setup of the CG solver and is only efficient for high order elements. The present work is based on a novel distributed FEM reformulation using belief propagation (BP) that eliminates the dependency on any sparse data-structures or algebraic operations; hence, attacking the root-cause of the problem.

* Corresponding author.
E-mail addresses: yousef.el-kurdi@mail.mcgill.ca (Y. El-Kurdi), mmehri@mit.edu (M.M. Dehnavi), warren.gross@mcgill.ca (W.J. Gross), dennis.giannacopoulos@mcgill.ca (D. Giannacopoulos).

[1] These authors contributed equally to this work.

http://dx.doi.org/10.1016/j.cpc.2015.03.019
0010-4655/

The belief propagation algorithm, as proposed by Pearl in [10], is a recursive message passing algorithm on graphical models that efficiently computes the marginal distribution of each variable node by sharing intermediate results. If the graph is a tree, then BP is guaranteed to converge to exact marginals. However, if the graph contains cycles, as typically the case in many practical applications, then BP takes an iterative form, referred to as Loopy Belief Propagation (LBP), which can be used to obtain an approximation for the marginals [10–14]. BP recently showed excellent empirical results in certain applications, such as machine learning, channel decoding, and computer vision [15–25]. Shental et al. [26] introduced the Gaussian BP algorithm as a parallel solver for a linear system of equations by modeling it as a pairwise graphical model. While the solver showed great promise for highly parallel computations on diagonally dominant matrices [27], it does not scale for large FEM matrices. It also fails to converge for high order FEM problems [28,29]. In addition, such a solver would still require assembling a large sparse data-structure.

The introduced Finite Element Gaussian Belief Propagation (FGaBP) algorithm and its multigrid variant, the FMGaBP algorithm, presented in [28,30,31], are distributed reformulations of the FEM that results in highly efficient parallel implementations. The algorithms provide a highly parallel approach to processing the FEM problem, element-by-element, based on distributed message communication and localized computations. This provides an algorithm amicable to different parallel computing architectures such as multicore CPUs and manycore GPUs.

In this work, we introduce new formulations for the FGaBP algorithm that better exploit its distributed nature. The new algorithms provide more efficient memory bandwidth utilization and considerably lower computational complexity; that is, reducing the local computational complexity from $O(n^3)$ to $O(n^2)$, where n is the rank of the local (element) FEM matrix. We verify the numerical results of the new formulation using the definite Helmholtz equation with a known solution. We also compare the new formulations with state-of-the-art open-source libraries such as deal.II [6] and Trilinos [8] on modern multicore CPUs. Implementation details of FMGaBP on GPUs are also presented in this work and its performance is compared to multicore CPUs.

The paper is organized as follows. In Section 2, a background on the FGaBP and FMGaBP algorithms is provided, which illustrates the algorithms and their key parallel features. In Section 3, we present the new formulations that reduce computation and communication costs of FGaBP. Section 4 presents implementation details on multicore and manycore architectures. Finally in Section 5, we present and discuss speedup results and close with concluding remarks.

2. Preliminary

In this section, an overview of the FGaBP and FMGaBP algorithms is provided; illustrating their distributed attributes, which will later be used to develop more efficient variants of the algorithms.

2.1. The FGaBP algorithm

In the following, the FGaBP algorithm is presented in three main stages. First, the FEM problem is transformed into a probabilistic inference problem. Second, a factor graph model of the FEM problem is created to facilitate the execution of a computational inference algorithm such as BP. Finally, the FGaBP update rules and algorithm is presented.

2.1.1. FEM as a variational inference

The variational form of the Helmholtz equation as discretized by the FEM is generally represented as follows [32,33]:

$$\mathcal{F}(U) = \sum_{s \in \mathcal{S}} \mathcal{F}_s(U_s) \tag{1}$$

where $\mathcal{S}$ is the set of all finite elements (local functions); U_s are the field unknowns for element s; and $\mathcal{F}_s$ is the energy-like contribution of each finite element. The local function $\mathcal{F}_s$ takes a quadratic form that can be shown as:

$$\mathcal{F}_s(U_s) = \frac{1}{2} U_s^T M_s U_s - B_s^T U \tag{2}$$

in which M_s is the element characteristic matrix with dimensions $n \times n$ where n is the number of Local element Degrees of Freedom (LDOF), and B_s is the element source vector.

Conventionally, the FEM solution is obtained by setting $\frac{\partial \mathcal{F}}{\partial U} = 0$, which results in a large and sparse linear system of equations presented as:

$$Au = b \tag{3}$$

where A is a large sparse matrix with dimensions $N \times N$; N is the number of Global Degrees of Freedom (GDOF) of the linear system; and b is the Right-Hand Side (RHS) vector. The linear system is typically solved using iterative solvers such as the Preconditioned Conjugate Gradient (PCG) method when A is Symmetric Positive Definite (SPD).

The FGaBP algorithm takes a different approach by directly minimizing the energy functional (1) using the BP inference algorithm. A variational inference formulation of FEM is created by modifying (1) as follows:

$$\mathcal{P}(U) = \exp[-\mathcal{F}] \tag{4}$$

$$= \frac{1}{Z} \prod_{s \in \mathcal{S}} \Psi_s(U_s) \tag{5}$$

where Z is a normalizing constant, and $\Psi_s(U_s)$ are local factor functions expressed as:

$$\Psi_s(U_s) = \exp\left[-\frac{1}{2} U_s^T M_s U_s + B_s^T U_s\right]. \tag{6}$$

Considering applications where M_s is SPD, the function Ψ_s, as in (6), takes the form of an unnormalized multivariate Gaussian distribution. In addition, it can be shown using convex analysis [16,34] that $\mathcal{P}$ is a valid multivariate Gaussian distribution functional of the joint Gaussian random variables U. The solution point to the original problem, which is the stationary point of the functional $\mathcal{F}$, can be restated as:

$$\arg\min_U \mathcal{F} = \arg\max_U \mathcal{P}. \tag{7}$$

Since the Gaussian probability $\mathcal{P}$ is maximized when $U = \mu$, where μ is the marginal mean vector of $\mathcal{P}$, the FEM problem can alternatively be solved by employing computational inference for finding the marginal means of U under the distribution $\mathcal{P}$. Hence BP inference algorithms will be employed to efficiently compute the marginal means of the random variables U.

2.1.2. FEM factor graph

Because $\mathcal{P}$ is directly derived from the FEM variational form, it is conveniently represented in a factored form as shown in (5). As a result, we can define a graphical model to facilitate the derivation of the BP update rules. One widely used graphical model is a Factor Graph (FG) [15], which is a bipartite graphical model that directly represents the factorization of $\mathcal{P}$. In our setting, we refer to such a FG as the FEM-FG. The FEM-FG, as shown in Fig. 1(b), includes