INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING, *SERIES B*
Volume 5, Number 1-2, Pages 123–135

RESEARCH OF MULTICORE-BASED PARALLEL GABP ALGORITHM WITH DYNAMIC LOAD-BALANCE

HANYUAN ZHENG, ANPING SONG, ZHIXIANG LIU, LEI XU, MINCHAO WANG, AND WU ZHANG*

Abstract. Based on Gaussian Belief Propagation(GaBP) algorithm for solving sparse symmetric linear equations, an iterative acceleration optimization method of GaBP is studied and a corresponding optimized storage scheme is proposed. We explore the parallelism and load balancing features of this algorithm and present a multicore-based parallel GaBP algorithm with dynamic load-balance. The numerical results indicate that this algorithm can solve large scale sparse symmetric linear equations with good results and high parallel efficiency.

Key words. multicore-based parallelization, GaBP algorithm, load balance, linear equations.

1. Introduction

Solving linear equations $Ax = b$ is the fundamental problems in various scientific and engineering computing. Numerical methods, such as finite element method, finite difference method, spectral method, and finite volume method [1, 2, 10, 3], convert the actual problem into the problem of solving sparse linear equations. With the increase of the scale and complexity of problems, how to effectively solve large scale sparse linear equations has been a hot area [4].

For solving sparse linear equations, iterative method is mainly used. The iterative method includes classical iterative method, such as Jacobi method, SOR method, Krylov subspace method which is very popular in recent years [5, 6]. In 2008, Ori Shental et al proposed an iterative method for symmetric diagonally dominant linear equations Gaussian Belief Propagation(GaBP) [7]. GaBP algorithm converts the problem of solving linear system into solving the problem of Probability and information dissemination which differs from the classical iterative method and Krylov subspace method. For symmetric diagonally dominant linear equations, GaBP algorithm has a good convergence, and is essentially equivalent to the classic Gauss elimination method.

The main objective of this paper is how to efficiently solve large scale sparse symmetric linear equations. We study the iterative acceleration method to optimize the GaBP algorithm based on its classical GaBP counterpart. By exploring the parallelism and features of GaBP algorithm, we present a multicore-based parallel GaBP algorithm with the feature of dynamic load-balance to solve large-scale sparse linear equations. Numerical experiment of solving large scale are fulfilled and results are compared with other algorithms.

The rest of the paper is organized as follows. In Section 2, GaBP Algorithm will be described in detail. the iterative acceleration optimization method of GaBP and a corresponding optimized storage scheme is shown in Section3. We discuss the experimental results in section 4. Finally, Section 5 concludes the paper.

Received by the editors January 1, 2014 and, in revised form, March 24, 2014.
2000 *Mathematics Subject Classification.* 65F10, 65F50, 68W10.
*Corresponding author.

2. GaBP Algorithm

In this section, we will review the classical GaBP algorithm [7, 8, 9]. For symmetric diagonally dominant linear equations

$$Ax = b, A \in \Re^{n\times n}, x, b \in \Re^n, \tag{1}$$

the coefficient matrix A is a nonsingular symmetric diagonally dominant matrix.

2.1. Symmetric Linear Equations and Its Probability Inference Model. First, We connect undirected graph with symmetric linear equations. Given an undirected graph $G = (V, E)$, where V is a set of all vertices in G corresponding to variables x in linear equations and E is the set of all edges associated with the non-zero elements in matrix A.

Now, we define the following joint Gaussian probability density function based on the coefficient matrix A and the observation vector b

$$p(x) \sim \exp(-\frac{1}{2}x^T Ax + b^T x), \tag{2}$$

and its corresponding graph G consisting of edge potentials ('compatibility functions') ψ_{ij} and self potentials ('evidence') ϕ_i. These graph potentials are simply determined according to the following pairwise factorization of the Gaussian function (2)

$$p(\mathbf{x}) \propto \prod_{i=1}^{n} \phi_i(x_i) \prod_{\{i,j\}} \psi_{ij}(x_i, x_j). \tag{3}$$

where $\psi_{ij}(x_i, x_j) \triangleq \exp(-x_i A_{ij} x_j)$ and $\phi_i(x_i) \triangleq \exp(-\frac{1}{2}A_{ii}x_i^2 + b_i x_i)$.

Proposition 1. *([7] Proposition 1 Solution and inference). The computation of the solution vector x^* is identical to the inference of the vector of marginal means $\mu = \mu_1, \cdots, \mu_n$ over the graph G with the associated joint Gaussian probability density function $p(x) \sim N(\mu \triangleq A^{-1}b, A^{-1})$.*

According to Proposition 1, we can translate the problem of solving the linear system (1) from the algebraic domain to the domain of probabilistic inference, see Figure 1.

Next, we will introduce the BP(Belief Propagation) algorithm. The set of graph nodes $N(i)$ denotes the set of all the nodes neighboring the ith node. The set $N(i) \setminus j$ excludes the node j from $N(i)$.

2.2. BP Algorithm. Belief propagation (BP) is equivalent to applying Pearls local message-passing algorithm [11], originally derived for exact inference in trees, to a general graph even if it contains cycles (loops). The excellent performance of BP in these applications may be attributed to the sparsity of the graphs.

The BP algorithm functions by passing real-valued messages across edges in the graph and consists of two computational rules, namely the $'$sum-product rule$'$ and the $'$product rule$'$. For a graph G composed of potentials ψ_{ij} and ϕ_i as previously defined, the conventional sum-product rule becomes an integral-product rule and the message $m_{ij}(x_j)$ [12], sent from node i to node j over their shared edge on the graph, is given by

$$m_{ij}(x_j) \propto \int_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{k\in N(i)\rfloor} m_{ki}(x_i)dx_i. \tag{4}$$