J. Parallel Distrib. Comput. 106 (2017) 62–78

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

The nine node Extrapolated Diffusion method for weighted torus graphs

K.A. Dimitrakopoulou *, N.M. Missirlis *

Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Panepistimiopolis, 157 84, Athens, Greece

HIGHLIGHTS

- A new Diffusion scheme (NEDF) for balancing the load in a torus network is proposed.
- NEDF uses the nine nearest nodes of a source node.
- Five alternative communication routings are studied.
- We find closed form formulae for the optimum values of the parameters.
- The cross communication routing improves the performance by 60%.

ARTICLE INFO

Article history:
Received 16 July 2015
Received in revised form
1 February 2017
Accepted 22 February 2017
Available online 21 March 2017

Keywords:
Laplacian matrix
Load balancing
Weighted torus
Iterative diffusion
Fourier analysis

ABSTRACT

The convergence analysis of the Extrapolated Diffusion (EDF) method was developed in Karagiorgos and Missirlis (2008) and Markomanolis and Missirlis (2010) for 2D weighted torus and mesh graphs, respectively using the set $\mathcal{N}_1(i)$ of the nearest neighbors of a source node i in the graph. In the present work we propose a Diffusion scheme which employs the set $\mathcal{N}_1(i) \cup \mathcal{N}_2(i)$, where $\mathcal{N}_2(i)$ denotes the set of the nearest neighbors of a source node i with path length two in an attempt to improve the performance of the new method. We develop the convergence analysis of the new EDF method by considering two subsets of $\mathcal{N}_2(i)$ for 2D weighted torus graphs. In particular, we study five different communication routes for computing the load of each node i in the torus graph and for each route we find closed form formulae for the optimum values of the edge weights, the extrapolation parameters and the convergence factor of the new EDF scheme. A comparison of the convergence factors of all these EDF schemes reveals a 60% improvement in the performance using the cross communication route compared to the conventional EDF method, a fact which is shown theoretically and experimentally.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

The efficient use of the available computational resources for running parallel applications in a distributed computing environment leads to the load balancing problem. Many parallel applications produce different workload during execution time. Parallel weather prediction simulations constitute typical applications that require load balancing techniques. In such applications a geometric space is discretized by a 3D-mesh. Then a domain decomposition technique assigns a subdomain to each processor of the network. Each processor performs the computations on mesh points in each subdomain independently. The involved computations in these meshes are of two kinds: "dynamics" and "physics". The dynamics computations correspond to the fluid dynamics of the atmosphere and are applied to the horizontal field. The physics computations represent the natural procedures such as clouds, moist convection, the planetary boundary layer and surface processes and are applied to the vertical level. The computations of a vertical column are local, that is, they do not need data from the neighboring columns and are implicit in nature. As the physics computations differ from one subdomain to the other the processors have an unbalanced load. In such situations the load has to be rebalanced.

Most of the existing iterative load balancing schemes [9] involve two steps. The first step calculates a balancing flow. This flow is used in the second step in which load balancing elements are migrated accordingly. This paper focuses on the first step. It is well

* Corresponding authors.
E-mail addresses: kdim@di.uoa.gr (K.A. Dimitrakopoulou), nmis@di.uoa.gr (N.M. Missirlis).
URLs: http://parallel.di.uoa.gr/PSCL/members/dimitrakopoulou.html (K.A. Dimitrakopoulou), http://parallel.di.uoa.gr/PSCL/members/missirlis.html (N.M. Missirlis).

http://dx.doi.org/10.1016/j.jpdc.2017.02.012
0743-7315/© 2017 Elsevier Inc. All rights reserved.

known that in distributed systems the communication complexity plays an important role. Therefore, a load balancing algorithm should have a minimum flow and use local communication. Iterative load balancing methods have these characteristics and this is the reason for the renewed interest in their study. Two main categories are the diffusion [1,3] and the dimension exchange [3,18] algorithms.

The performance of a balancing algorithm can be measured in terms of number of iterations it requires to reach a balanced state and in terms of the amount of load moved over the edges of the underlying processor graph. In [4] it is shown that all local iterative diffusion schemes calculate a minimal flow with respect to a weighted l_2-norm. In the Diffusion (DF) method [3,1] a processor simultaneously sends workload to its neighbors with lighter workload and receives from its neighbors with heavier workload. It is assumed that the system is synchronous, homogeneous and the network connections are of unbounded capacity. Under the synchronous assumption the DF method has been proved to converge in polynomial time for any initial workload [1]. More specifically DF is given by the following iterative scheme

$$u_i^{(n+1)} = u_i^{(n)} - \sum_{j\in\mathcal{N}_1(i)} c_{ij}(u_i^{(n)} - u_j^{(n)}), \quad n = 0, 1, 2, \ldots, \tag{1}$$

where $c_{ij} \in \mathbb{R}^+$ are the edge weights, $\mathcal{N}_1(i)$ is the set of the nearest neighbors of node i of the graph $G = (V, E)$ and $u_i^{(n)}$, $i = 0, 1, 2, \ldots, |V|$ is the load after the nth iteration on node i. The main problem here is the determination of the parameters c_{ij} such that the rate of convergence of DF is maximized. However, this is an optimization problem with multiple parameters. Indeed, one has to find a set of values for c_{ij}'s which minimize the convergence factor γ of DF. Initially, Cybenko [3] suggested choosing $c_{ij} = 1/(1 + d(i))$, where $d(i) = |N(i)|$ and proved that for binary n-cubes this is an optimal choice. Similar convergence results were also obtained in [1,8]. A special case of the problem was solved by Xu and Lau [18] assuming all edge weights are equal to a single parameter and determined optimal values for the cases of the N-ary n-cube and its variant, the nD-torus using circulant matrix theory. A first attempt to find optimal values for c_{ij}'s, using semi-definite programming, was made in [5] where optimal numerical values for edge weights of certain graphs with small cardinality were computed. A second step towards this direction using results of Cartesian product of graphs was [7]. However, the value of the involved parameter, which optimizes the rate of convergence of the DF method, is computed numerically resulting in a significant increase of time complexity. More specifically, the optimum value of the involved parameter requires the determination of the maximum and minimum eigenvalue of the weighted Laplacian matrix in a way similar to [17]. But now the weighted Laplacian is not a circulant matrix and the relevant theory does not apply. To circumvent this problem we use Fourier analysis [2] in a similar way as to the study of stability in partial differential equations. This approach is straight forward and can be easily applied to torus, n-cubes and mesh graphs [11,14,10].

In [11] the following Extrapolated Diffusion (EDF) method was introduced

$$u_i^{(n+1)} = u_i^{(n)} - \tau \sum_{j\in\mathcal{N}_1(i)} c_{ij}(u_i^{(n)} - u_j^{(n)}), \quad n = 0, 1, 2, \ldots, \tag{2}$$

where $\tau \in \mathbb{R} \setminus \{0\}$ is the extrapolation parameter. In fact, we let the extrapolation parameter vary for each node i, so instead of τ in (2) we introduced a set of parameters τ_i, $i = 0, 1, 2, \ldots, |V|$ and referred to (2) as the local EDF method [11,14]. Note that if $\tau = 1$, (2) is the Diffusion (DF) method [3,1]. The generalized problem of determining optimum values for the edge weights c_{ij} and τ_i was first solved in [11]. In particular, closed form formulae for the optimum values for τ_is and c_{ij}s were determined for the nD-torus [11] and nD-mesh [14] by applying local Fourier analysis [2,12]. These studies showed that for square tori and meshes EDF coincides with DF whereas for orthogonal tori and meshes it becomes twice as fast as DF. In addition, it was shown that EDF for orthogonal tori is four times faster compared to orthogonal meshes [14].

In the present work we study the case of increasing the number of neighbors of a node in the (i, j) position of a 2D torus weighted graph for the computation of its load. In the local EDF method we use the load of the four nearest neighbors north, south, east and west for the computation of the load of a node (i, j). In an attempt to improve the performance of the local EDF method we add more local information a fact which is achieved by adding the load of the nearest nodes with path length two from a source node (i, j). In particular, we consider two different subsets of neighbors with path length two and study the impact of five alternative communication routes of load transfer on the rate of convergence of the new local EDF method which will be referred as the local Nine node EDF (NEDF) method since it uses the load of nine nodes in total. This technique is used in the discretization of mixed derivatives or derivatives of high dimension in the Partial Differential equations [12,13]. For each communication route we find closed formulae for the optimum values of the parameters τ_{ij} such that the convergence of the local NEDF method is maximized. This is a significant reduction in the time complexity compared to the numerical determination of τ_{ij}. We also determine the optimum values of the edge weights c_{ij} in each case. As the nine node Laplacian is not a circulant matrix we use local Fourier analysis. A comparison of all the different communication routes reveals that the cross route produces the maximum rate of convergence of the NEDF method which is improved asymptotically by 3.2 times compared to the local EDF method. A final innovative characteristic of our approach is the formulation of the study of convergence of the local NEDF method as an optimization problem with constraints.

The rest of the paper is organized as follows. In Section 2 we introduce the general form of the local NEDF method for a graph $G = (V, E)$ using the neighbors with path length one from a source node i as well as the set of neighbor nodes with path length two. Next, we focus on the 2D weighted torus graph and in Section 3 we study the cross communication route for the load transfer by taking into account not only the load of the four nearest neighbor nodes but also of the four additional nodes which lie north, south, east and west with path length two from a source node (i, j). In the next four sections we consider alternative communication routes by taking into account the load of the four additional nodes which lie at the four corners of the square with center at a source node (i, j) and determine optimum values for the extrapolation parameters and the edge weights. In Section 4 we study the horizontal–vertical route, which first transfers the load of the neighbor nodes in the horizontal and then in the vertical direction. The case where this load transfer is made first in the vertical and then in the horizontal direction is studied in Section 5. The tetra-gammadion route and its mirror image, the anti tetra-gammadion route, are studied in Sections 6 and 7, respectively. Section 8 presents our numerical results and conclusions.

2. The Nine node Extrapolated Diffusion (NEDF) method

Let $G = (V, E)$ be a connected, weighted, undirected graph with $|V|$ nodes and $|E|$ edges. Let $c_{ij} \in \mathbb{R}^+$ be the weight of edge $e_{ij} \in E$, $u_i \in \mathbb{R}$ be the load of node $v_i \in V$ and $u \in \mathbb{R}^{|V|}$ be the vector of load values. Further, we consider the following iterative scheme that requires communication not only with adjacent nodes but also with their neighbors

$$u_i^{(n+1)} = u_i^{(n)} - \tau \sum_{j\in\mathcal{N}(i)} c_{ij}(u_i^{(n)} - u_j^{(n)}), \tag{3}$$

Download English Version:

https://daneshyari.com/en/article/4951548

Download Persian Version:

https://daneshyari.com/article/4951548

Daneshyari.com