Nikolaos, N M Missirlis

The nine neighbor Extrapolated Diffusion

Department of Informatics and Telecommunications
University of Athens
Panepistimiopolis
15784
Ilisia
Athens
Greece
nmis@di.uoa.gr
Aikaterini Dimitrakopoulou

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 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. More specifically DF is given by the following iterative scheme

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

where $c_{ij} > 0$ are the edge weights (or diffusion parameters), $N(i)$ 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. Various attempts to solve this problem are presented in [3, 6, 4, 5] and [9]. In [7] 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 $\tau \in \mathbb{R} \setminus \{0\}$ and $c_{ij} > 0$ are the extrapolation parameter and the edge weights, respectively, $\mathcal{N}_1(i)$ is the set of the nearest neighbors of the 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. In fact, we let the extrapolation parameter vary for each node i, so instead of τ in (2) we introduced a set of parameters $\tau_i, i = 0, 1, 2, \ldots, |V|$ and called (2) the local EDF method [7, 8]. Note that if $\tau = 1$, (2) is the classical Diffusion (DF) method [3, 1]. In [9] optimum values for the edge weights c_{ij} were determined under the hypothesis that they are all equal. The generalized problem of determining optimum values for the edge weights c_{ij} and τ_i was first solved in [7]. In particular, closed form formulae for the optimum values for τ_is and c_{ij}s were determined for the nD-torus [7] and nD-mesh [8] by applying local Fourier analysis [2]. As a result, it was found 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 than orthogonal meshes [8].

In the present work we consider the case of increasing the number of neighbors of node i for the computation of its load. In other words we consider the nine neighbor weighted Laplacian matrix formed not only by the five nearest neighbor nodes but also by their four neighbors with path length two from node i in an attempt to increase the convergence rate of the EDF method. As a consequence, we show that the rate of convergence of EDF is improved asymptotically by 60% for torus graphs.

The rest of the paper is organized as follows: In section 2 we introduce the local NEDF method using nine neighbors of node i. In section 3 we find closed form formulae for the optimum values of the parameters τ_i such that the convergence of the local NEDF method is maximized. The determination of the optimum values for the edge weights c_{ij} is presented in section 4 with a comparison of the NEDF and EDF methods with five and nine neighbors, respectively for torus graphs. Section 5 presents our numerical experiments and finally, in section 6 we state our conclusions.

Bibliography

[1] J. E. Boillat, Load balancing and poisson equation in a graph, Concurrency: Practice and Experience, 2, 289-313, 1990.

[2] A. Brandt, Multi-level adaptive solutions to boundary-value problems, Math. Comput. 31, 333-390,1977,

[3] G. Cybenko, Dynamic load balancing for distributed memory multiprocessors, J. Parallel and Distr. Comp., 7, 1989, 279-301.

[4] R. Diekmann, S. Muthukrishnan, M. V. Nayakkankuppam, Engineering diffusive load balancing algorithms using experiments. LNCS 1253, 111-122, 1997.

[5] R. Elsässer, B. Monien and R. Preis, Optimal Diffusion schemes and Load Balancing on Product Graphs, Parallel Proc. Letters, 14, 61-73, 2004.

[6] J. Hong, X. Tau and M. Cheu, From local to global: an analysis of nearest neighbor balancing on hypercube, in Proc ACM Symp. on SIGMETRICS, 73-82, 1988.

[7] G. Karagiorgos and N. Missirlis, Convergence of the diffusion method for weighted torus graphs using Fourier analysis, J. Theor. Comp. Sc., 401 (1-3)1-16,2008.

[8] G. S. Markomanolis and N. M. Missirlis,Optimum Diffusion for Load Balancing in mesh networks,Euro-Par 2010,230-241,2010.

[9] C.Xu and F. M. Lau, Load balancing in parallel computers: Theory and Practice

[10] D. M. Young, Iterative Solution of Large Linear Systems, Academic, 2003.