Journal of Computer and System Sciences 81 (2015) 159–185

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Randomized diffusion for indivisible loads

Petra Berenbrink [a], Colin Cooper [b], Tom Friedetzky [c], Tobias Friedrich [d,*], Thomas Sauerwald [e]

[a] Simon Fraser University, Burnaby, Canada
[b] King's College London, UK
[c] Durham University, UK
[d] Friedrich-Schiller-Universität Jena, Germany
[e] University of Cambridge, UK

ARTICLE INFO

Article history:
Received 3 December 2010
Received in revised form 31 March 2014
Accepted 17 April 2014
Available online 5 June 2014

Keywords:
Distributed computing
Load balancing
Diffusion
Randomized algorithms
Random walk

ABSTRACT

We present a new randomized diffusion-based algorithm for balancing indivisible tasks (tokens) on a network. Our aim is to minimize the discrepancy between the maximum and minimum load. The algorithm works as follows. Every vertex distributes its tokens as evenly as possible among its neighbors and itself. If this is not possible without splitting some tokens, the vertex redistributes its excess tokens among all its neighbors randomly (without replacement). In this paper we prove several upper bounds on the load discrepancy for general networks. These bounds depend on some expansion properties of the network, that is, the second largest eigenvalue, and a novel measure which we refer to as refined local divergence. We then apply these general bounds to obtain results for some specific networks. For constant-degree expanders and torus graphs, these yield exponential improvements on the discrepancy bounds. For hypercubes we obtain a polynomial improvement.

1. Introduction

During the last years, large parallel networks became widely available for industrial and academic users. An important prerequisite for their efficient usage is to balance their work efficiently. Load balancing is also known to have applications to scheduling, routing, numerical computation, and finite element computations.

In this paper we analyze a very simple neighborhood-based load balancing algorithm. We assume that the processors are connected by an arbitrary d-regular network. In the beginning, every vertex has a certain number of tokens (load) and the goal is to distribute the tokens as evenly as possible. More precisely, we aim at minimizing the difference between the minimum and maximum load, which we call *discrepancy*.

Neighborhood-based load balancing algorithms normally operate in parallel steps. In each step, every processor is allowed to probe the load of all of its neighbors (*diffusion load balancing*), or to probe the load of one neighbor (*dimension exchange*). Then each processor has to decide how much load it will forward to the neighbors in question. Here we consider a very natural diffusion-based approach where every processor tries to balance the load locally. This means that along each edge, a load of *load-difference*$/(d+1)$ is sent to the vertex with less load. This is exactly the approach in the *continuous* diffusion model where tokens can be split arbitrarily. This method balances the load perfectly.

* Corresponding author.

http://dx.doi.org/10.1016/j.jcss.2014.04.027

In contrast to continuous diffusion, we consider the (arguably more realistic [25]) case of *discrete* diffusion where tokens are indivisible. Quantifying by how much the integrality assumption decreases the efficiency of load balancing is an interesting question and has been posed by many authors (e.g., [12,16,20,23–25]).

In the common *edge-oriented* view of e.g. [13,15,24], for each edge one has to decide between transferring either $\lceil \textit{load-difference}/(d+1) \rceil$ or $\lfloor \textit{load-difference}/(d+1) \rfloor$ tokens (referred to as rounding up or rounding down). Rounding up results in a load balancing algorithm that keeps sending tokens back and forth between processors with a small load difference. Another disadvantage is that the approach can generate "negative loads" for vertices with only a few tokens. On the other hand, always rounding down cannot balance better than $d \cdot \mathrm{diam}(G)$, where $\mathrm{diam}(G)$ denotes the diameter of the underlying graph G. To overcome these problems we adopt a *vertex-oriented* view in this paper. We present a randomized diffusion load balancing algorithm where the vertices (not edges) decide randomly how much they are sending.

1.1. Related work

Due to the vast amount of literature on load balancing, we consider only previous work dealing with diffusion load balancing, or randomized algorithms for neighborhood-based load balancing. We do not consider the dimension exchange model in general, or token distribution model where only one token can be sent to a neighbor per step.

Continuous diffusion. The diffusion model was first studied by Cybenko [5] and, independently, Boillat [3]. Cybenko [5] (see also [23,25]) shows a tight connection between the convergence rate of the diffusion algorithm and the absolute value of the second largest eigenvalue $\lambda_{\max}$ of the diffusion matrix $\mathbf{P}$ with $\mathbf{P}_{ij} = 1/(d+1)$ if $\{i, j\} \in E$. Subramanian and Scherson [25] observe similar relations between convergence time and certain properties of the underlying network like electrical and fluid conductance.

Muthukrishnan et al. [23] refer to the above diffusion model as the *first order scheme* and generalize it to the so called *second order scheme*. Here the load transferred over an edge (i, j) in step t does not only depend on the load difference of i and j, but also on the amount of load transferred over the edge in step $t-1$. Diekmann et al. [7] extend the idea of [23] and propose a general framework to analyze the convergence behavior of a wide range of diffusion type methods.

Discrete diffusion. In order to approximate the idealized process by a discrete process with indivisible load, Rabani et al. [24] consider a diffusion algorithm (called RSW algorithm in the following) which always rounds down the indivisible load on each edge. To quantify the deviation of the discrete load from the idealized process, they propose a natural measure, the *local divergence* Ψ_1. The local divergence measures the sum of load differences across all edges in the network, aggregated over time. They give a general bound on the divergence in terms of $\lambda_{\max}$, which denotes the absolute value of the second largest eigenvalue of the diffusion matrix $\mathbf{P}$. By a more careful analysis, they also get an improved upper bound on Ψ_1 for tori, resulting in a tight bound on the discrepancy achieved by their algorithm.

Discrete load balancing via random walks. Elsässer et al. [10–12] proposed an algorithm based on random walks. They show that after $\mathcal{O}(\log(Kn)/(1-\lambda_{\max}))$ steps, the maximum load is at most the average load plus a constant [11]. In comparison to our algorithm, their algorithm is more complicated and different from the usual diffusion framework. For example, vertices require an estimate of n and have to compute the average load during the balancing procedure. Moreover, the final stage uses concurrent random walks (representing tokens) to reduce the maximum load. In this stage, the load transfer along an edge may be much smaller (or higher) than $\textit{load-difference}/(d+1)$.

Discrete neighborhood load balancing with randomization. In [13] the last two authors analyze a randomized version of the dimension-exchange algorithm using randomly generated or deterministic matchings. In their algorithm, the decision to round up or down is randomized. For detailed results see Table 1. Note that in their case every node exchanges load with at most one neighbor. This is typically much easier to analyze than diffusion algorithms.

Friedrich et al. [15] analyze a deterministic modification of the standard diffusion algorithm for hypercubes and constant-dimensional tori. The idea is that each edge keeps tracks of its own rounding errors. In each step an edge's decision to round up or down is done such that the sum of its rounding errors is minimized. Again, the detailed results can be found in Table 1. Friedrich et al. [15] also consider a randomized version of the diffusion algorithm. Their approach is edge-based. Edges decide independently at random whether to round up or down. The probabilities are chosen such that, in expectation, the behavior of the continuous diffusion algorithm is mimicked. They present a general upper bound for their approach in terms of $\lambda_{\max}$. Note that both algorithms in [15] may generate negative load due to the edge-based rounding.

Source of inspiration. We wish to point out that our work was inspired by recent combinatorial results regarding so-called *rotor-router walks* [4,8]. Unlike in a random walk, in a rotor-router walk each vertex serves its neighbors in a fixed order. The resulting (completely deterministic) walk nevertheless closely resembles a random walk in several respects. Similarly, one can say that in each round of our load-balancing algorithm a vertex chooses a random order of its neighbors (and itself) and sends around all its tokens in this order in a round-robin fashion.

1.2. Our contribution

Algorithm. We consider a vertex-based randomized diffusion algorithm for the discrete model with indivisible tokens. Let d be the degree of the (regular) network and let X_i be the load of vertex i. Our algorithm works as follows. First, vertex i sends $\lfloor X_i/(d+1) \rfloor$ tokens to each neighbor and keeps the same amount of tokens for itself. Then the remaining