Tight & Simple Load Balancing

Petra Berenbrink
Universität Hamburg
Hamburg, Germany
petra.berenbrink@uni-hamburg.de

Tom Friedetzky
Durham University
Durham, U.K.
tom.friedetzky@dur.ac.uk

Dominik Kaaser
Universität Hamburg
Hamburg, Germany
dominik.kaaser@uni-hamburg.de

Peter Kling
Universität Hamburg
Hamburg, Germany
peter.kling@uni-hamburg.de

***Abstract*—We consider the following load balancing process for m tokens distributed arbitrarily among n nodes connected by a complete graph: In each time step a pair of nodes is selected uniformly at random. Let ℓ_1 and ℓ_2 be their respective number of tokens. The two nodes exchange tokens such that they have $\lceil(\ell_1+\ell_2)/2\rceil$ and $\lfloor(\ell_1+\ell_2)/2\rfloor$ tokens, respectively. We provide a simple analysis showing that this process reaches almost perfect balance within $\mathrm{O}(n\log n + n\log\Delta)$ steps, where Δ is the maximal initial load difference between any two nodes. This bound is asymptotically tight.**

I. Introduction

We consider a discrete load balancing problem for m tokens on n identical resources. Each node starts with an arbitrary number of tokens and the objective is to distribute the tokens as evenly as possible. In every step, a pair of resources is chosen uniformly at random and their loads (number of tokens) are balanced as evenly as possible. The continuous case, where tokens can be arbitrarily divided and the resources can balance their number of tokens evenly, is well understood. In this paper we consider the discrete case where tokens are non-divisible. Hence, if one of the resources has load ℓ and the other one ℓ', the load will be divided as $\lceil \ell+\ell'/2\rceil$ and $\lfloor \ell+\ell'/2\rfloor$. We provide a simple and elementary proof that this process takes, w.h.p. (with high probability[1]), $\mathrm{O}(n\log n + n\log\Delta)$ time steps to reach almost perfect balance (see Section II). Here, Δ is the maximal initial load difference between any two nodes[2] and almost perfect balance means that all nodes have a load in $\{\lfloor\varnothing\rceil-1, \lfloor\varnothing\rceil, \lfloor\varnothing\rceil+1\}$, where $\varnothing = m/n$ and $\lfloor\varnothing\rceil$ is $\varnothing$ rounded to the nearest integer. Our bound is asymptotically tight (see discussion after Theorem 1). We also provide an empirical study that not only confirms our theoretical results but indicates that the constants hidden by the asymptotic analysis are small (see Section III).

We should first like to note that the process we analyze has been considered before. In [17] the authors consider the process in a more general setting where the nodes are connected by a graph and several balancing actions can be performed in parallel (but only one per node). However their results, applied to our setting, are not as tight as ours; they show a maximum load of $\lfloor\varnothing\rceil + c$ for a suitably chosen constant c. In [14] the authors consider the *population model*, with our process being used within an algorithm that calculates the proportion of players holding one of two distinct opinions.

The first part of our analysis is based on a standard potential function argument. We then switch from a resource-based reasoning to a token-based reasoning, where we assign a height to each token held by a node. This height is used to move tokens in a certain order, thereby simplifying the analysis considerably. The process itself, however, is blissfully unaware of this, and coupling the actual with the analyzed process is trivial. We believe this approach to analyzing this (kind of) process may be useful elsewhere and hence be of independent interest.

A. Related Work

There is a vast body of literature on iterative load balancing, even when considering theoretical results only. Many of the results for iterative load balancing are for the continuous case, where load items can be broken into arbitrarily small pieces. As it is beyond the scope of this article to provide a complete survey, here we focus on results for discrete load balancing only. For results about continuous load balancing see, for example, [6, 12]. There are also many results in the context of balancing schemes where not the resources try to balance their load but the tokens (acting as selfish players) try to find a resource with minimum load. See [7] for a comprehensive survey and [1, 5, 11] for some recent results.

A vast majority of the results are for a more general model where the resources are represented by the nodes of a graph and the resources can balance with all or a subset of the neighboring resources only. Here one distinguishes between diffusion load balancing and the dimension exchange (or matching) model. In diffusion load balancing nodes can, in every step, balance their load with all neighbors, whereas in the matching model the edges which are used for load balancing form a matching. In the latter model every resource is only involved in one balancing action per step which makes the analysis much easier. Results for general graphs are usually expressed as a function of graph parameters like the second largest eigenvalue. In this overview we will discuss the results for general graphs but the detailed performance results will be stated for complete graphs only. Note that, for complete graphs, our results easily carry over to the dimension exchange model with randomly chosen matchings ([4, 17]).

The authors of [15] proved the first rigorous result for the discrete load balancing in the diffusion model. They assume

[1]The expression *with high probability* refers to a probability of $1-n^{-\Omega(1)}$.

[2]We assume $\Delta > 0$ (such that $\log\Delta$ is well defined); otherwise the system is already perfectly balanced.

that the number of tokens sent along each edge is obtained by rounding down the amount of load that would be sent in the continuous case. Using this approach, they established that the discrepancy (difference between minimum and maximum load) is at most $\mathrm{O}(n^2)$ after $\mathrm{O}(\log(Kn))$ steps, where K is the initial discrepancy. Similar results for the matching model were shown in [9]. In [16] the authors show results for a wide class of diffusion and matching load balancing protocols. They introduce the so-called local divergence, which is a natural parameter that essentially aggregates the sum of load differences over all edges in all rounds and prove that the local divergence yields an upper bound on the maximum deviation between the continuous and discrete case of a protocol. To translate a continuous into a discrete protocol, they assume that nodes round down the amount of tokens which is send over an edge. For complete graphs they show a discrepancy of $\mathrm{O}(n \log n)$ after $\mathrm{O}(\log(Kn))$ steps. Note that all the above results are for general graphs.

While always rounding down may lead to quick stabilization, the discrepancy tends to be quite large, a function of the diameter of the graph. Therefore, the authors of [16] suggested to use randomized rounding in order to get a better approximation of the continuous case (rounding is not necessary for complete graphs). In [8] the authors show several results for a randomized protocol in the matching model. For complete graphs their result shows a discrepancy of $\mathrm{O}\left(n\sqrt{(\log n)}\right)$ after $\mathrm{O}(\log(Kn))$ steps. Later, Berenbrink et al. [3] extended some of these results to the diffusion model. In [17] improve these results showing, for the first time, a constant discrepancy. For complete graphs the balancing time is again $\mathrm{O}(\log(Kn))$ steps. Note that, to compare their results to ours, one has to divide our run time by n (every n steps of our protocol generates a random matching of size $\mathrm{O}(n)$).

In [4] the authors propose a very simple potential function technique to analyze discrete diffusion load balancing schemes, both for discrete and continuous settings. They sequentialize the load balancing actions of the diffusion approach in a suitable way, and then we show that the potential decreases after each of these sequential load balancing actions. They apply their approach to a parallel version of our setting where every node randomly chooses a load balancing partners from among the set of all other nodes. The balancing time is again $\mathrm{O}(\log(Kn))$ and the maximum discrepancy $\mathrm{O}(\sqrt{n})$. In [2] the authors show another approach that turns any continuous into a discrete algorithm. For complete graphs the approach yields a maximum discrepancy of $\mathrm{O}(n)$.

Another related strain of literature considers discrete, sequential load balancing, but with the restriction that only one token can move per time step. Goldberg [10] considered a simple local search process in this scenario: Tokens are activated by an independent exponential clock of rate 1. Upon activation, a token samples a random node and moves there if that node's load is smaller than the load at the token's current host node. It has recently been proved [5] that this process reaches perfect balance in $\mathrm{O}\left(\log n + \log(n) \cdot n^2/m\right)$ time (both in expectation and w.h.p.), which is asymptotically tight.

B. Model and Notation

Assume m indistinguishable tokens are distributed arbitrarily among n nodes of a complete graph. Define the *load vector* $\boldsymbol{L}(t) = (\ell_1(t), \ldots, \ell_n(t)) \in \mathbb{Z}^n$ at time t, where $\ell_i(t)$ is the number of tokens (load) assigned to node i at time t. The *discrepancy* $\Delta\boldsymbol{L}(t)$ at time t is the maximal load difference between any two nodes. Let $\Delta = \Delta\boldsymbol{L}(0)$ be the *initial discrepancy*. We define $\varnothing = m/n$ as the *average load* and use $\lfloor\varnothing\rceil$ to denote the average load rounded to the nearest integer.

Given the load vector $\boldsymbol{L}(t)$ at time t, our load balancing process performs the following actions during time step t: a) Two nodes u and v are selected uniformly at random without replacement. b) Their loads are updated according to $\ell_u(t+1) = \lceil(\ell_u(t) + \ell_v(t))/2\rceil$ and $\ell_v(t+1) = \lfloor(\ell_u(t) + \ell_v(t))/2\rfloor$.

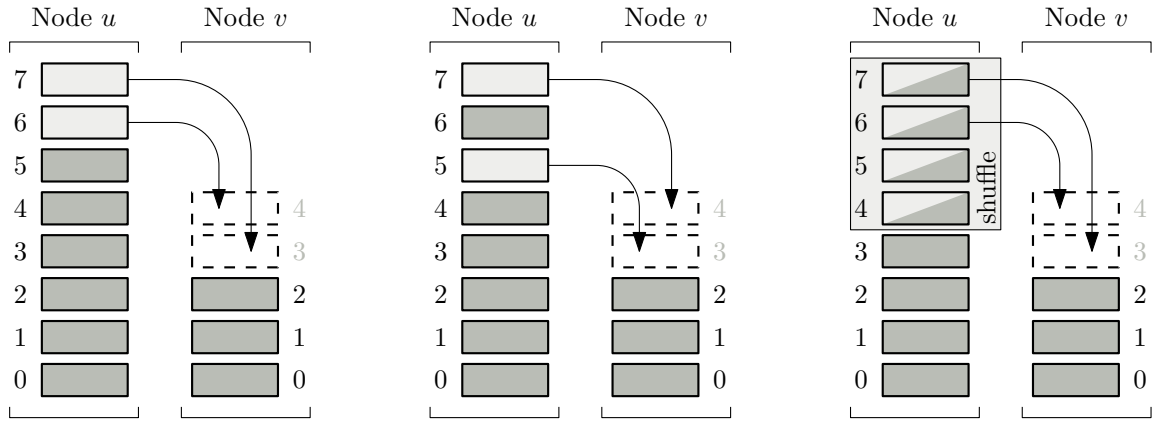
For the sake of the analysis we assume that tokens are ordered (arbitrarily) on each node. Based on this order, we define the *height* $h_b(t)$ of a token b at time t as the number of tokens that precede b in this order. The *normalized height* $\hat{\mathrm{h}}_b(t) = \mathrm{h}_b(t) - \lfloor\varnothing\rceil$ enumerates the tokens relative to the rounded average $\lfloor\varnothing\rceil$. Furthermore, we initially assume that balancing operations between two nodes operate in *stack mode*, where the topmost tokens of the node with higher load are moved to the node with lower load (see Figure 1a). For the second part of our analysis (Phase 2) we assume that balancing operations operate in *skip mode*, where every second token is moved (see Figure 1b). Finally, in the third part of our analysis (Phase 3), we assume that the excess tokens are first shuffled before the balancing operates in stack mode (see Figure 1c). Note that the mode does not influence the balancing process but merely facilitates the analysis.

II. Analysis

We split the analysis into three phases. In Phase 1 we use a potential function argument to show that, w.h.p., it takes $\mathrm{O}(n \log n + n \log \Delta)$ time steps until at most $n/2$ nodes have a load larger than $\varnothing + \Theta(1)$. In Phase 2 we look at individual tokens and prove that, w.h.p., it takes $\mathrm{O}(n \log n)$ more time steps until all nodes have load at most $\varnothing + \Theta(1)$. Finally, in Phase 3 we prove that, w.h.p., it takes $\mathrm{O}(n \log n)$ further time steps until the maximum load is at most $\lfloor\varnothing\rceil + 1$. Using a symmetry-based argument we get a similar bound on the minimum load and, thus, the following theorem.

Theorem 1. *Let $\boldsymbol{L}(0) \in \mathbb{N}_0^n$ be the initial load vector of the load balancing process on n nodes and let $\Delta = \Delta\boldsymbol{L}(0)$ be the initial discrepancy. Let furthermore T be the first time when all nodes have load in $\{\lfloor\varnothing\rceil - 1, \lfloor\varnothing\rceil, \lfloor\varnothing\rceil + 1\}$. With high probability, $T = \mathrm{O}(n \log \Delta + n \log n)$.*

Observe that Theorem 1 is tight: If $\Delta = \mathrm{poly}(n)$, with constant probability there are nodes that are not selected at all during the first $\mathrm{o}(n \log n)$ time steps. Otherwise, if Δ is superpolynomial in n, define $\beta = (m - \Delta)/n$ and consider the initial configuration where $\ell_1(0) = \beta + \Delta$ and



(a) Stack Mode (b) Skip Mode (c) Shuffle Mode

Fig. 1: Illustration of the different modes assumed for balancing operation during the analysis.

$\ell_i(0) = \beta$ for all $i \in \{2, 3, \dots, n\}$. The probability that node 1 takes part in any given interaction is $2/n$. Thus, during the first $(n \cdot \log \Delta)/4$ interactions, node 1 is activated $(\log \Delta)/2$ times in expectation and at most $(\log \Delta) \cdot 3/4$ times w.h.p. (by Chernoff, see Appendix A). If ℓ_1 denotes the load of node 1 during any such interaction, it looses at most $\lceil (\ell_1 - \beta)/2 \rceil$ tokens. If Δ is a power of two, we get that, after the first $(n \cdot \log \Delta)/4$ interactions, node 1 has, w.h.p., at least $\beta + \Delta/2^{(\log \Delta) \cdot 3/4} = \beta + \Delta^{1/4} = m/n + \omega(1)$ tokens, yielding a discrepancy of $\omega(1)$.

A. Phase 1: Potential Function Analysis

We analyze the process with the potential function

$$\Phi(\boldsymbol{\ell}) = \sum_{i=1}^{n} (\ell_i - \varnothing)^2 \tag{1}$$

for a load vector $\boldsymbol{\ell} \in \mathbb{N}_0^n$.

Lemma 1. *Let T_1 be the first time step for which $\Phi(\boldsymbol{L}(T_1)) < n$. W.h.p., $T_1 = \mathrm{O}(n \log n + n \log \Delta)$.*

Proof. We start by analyzing the expected change of the potential during one time step. Let $\delta(\boldsymbol{\ell}, i, j)$ be the potential drop of a fixed load vector $\boldsymbol{\ell} = (\ell_1, \dots, \ell_n) \in \mathbb{N}_0^n$ when nodes i and j are balancing. Then

$$\begin{aligned} \delta(\boldsymbol{\ell}, i, j) = (\ell_i - \varnothing)^2 + (\ell_j - \varnothing)^2 \\ - \left(\left\lceil \frac{\ell_i + \ell_j}{2} \right\rceil - \varnothing \right)^2 \\ - \left(\left\lfloor \frac{\ell_i + \ell_j}{2} \right\rfloor - \varnothing \right)^2 . \end{aligned} \tag{2}$$

We define the discretization error $r_\ell(i, j)$ as 1 if $\ell_i + \ell_j$ is odd and 0 otherwise. This allows us to expand and simplify the above expression to get

$$\begin{aligned} \delta(\boldsymbol{\ell}, i, j) &= \frac{\ell_i^2}{2} + \frac{\ell_j^2}{2} - \ell_i \ell_j - \frac{r_\ell(i,j)^2}{2} \\ &= \frac{(\ell_i - \ell_j)^2 - r_\ell(i,j)^2}{2} \\ &\geq \frac{(\ell_i - \ell_j)^2}{2} - 1/2 \ . \end{aligned} \tag{3}$$

Equation (3) implies that the potential never increases when two nodes balance (the only negative term is $-r_\ell(i,j)^2/2$, but $r_\ell(i,j) = 1$ implies $\ell_i \neq \ell_j$ and, thus, $(\ell_i - \ell_j)^2 \geq 1$). We now calculate the expected potential after one time step. Each pair of nodes is chosen uniformly at random with probability $1/\binom{n}{2}$. When chosen, the potential drops by $\delta(\boldsymbol{\ell}(t), i, j)$. Therefore,

$$\begin{aligned} &\mathbb{E}[\Phi(\boldsymbol{L}(t+1)) \mid \boldsymbol{L}(t) = \boldsymbol{\ell}] \\ &\quad = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{\binom{n}{2}} \cdot (\Phi(\boldsymbol{\ell}) - \delta(\boldsymbol{\ell}, i, j)) \\ &\quad \leq \Phi(\boldsymbol{\ell}) - \frac{1}{2\binom{n}{2}} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (\ell_i - \ell_j)^2 + \frac{1}{2} \ . \end{aligned} \tag{4}$$

We now use

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} (\ell_i - \ell_j)^2 = n \cdot \Phi(\boldsymbol{\ell}) \tag{5}$$

and obtain

$$\begin{aligned} &\mathbb{E}[\Phi(\boldsymbol{L}(t+1)) \mid \boldsymbol{L}(t) = \boldsymbol{\ell}] \\ &\quad \leq \Phi(\boldsymbol{\ell}) - \frac{1}{2\binom{n}{2}} \cdot n \cdot \Phi(\boldsymbol{\ell}) + \frac{1}{2} \\ &\quad \leq \left(1 - \frac{1}{n}\right) \cdot \Phi(\boldsymbol{\ell}) + \frac{1}{2} \ . \end{aligned} \tag{6}$$

Note that Equation (5) can be verified straightforwardly by expanding both sided. For completeness, the full calculations are given in Appendix B.

We now partition the time horizon into *rounds* of n consecutive time steps each and look at *successful* rounds

(in which the potential drops sufficiently). We then argue that $O(\log(\Phi(\boldsymbol{L}(0))/n))$ successful rounds suffice for the potential to drop below n and that, w.h.p., we have sufficiently many successful rounds among the first $O(\log n + \log \Delta)$ rounds.

Let round r consist of the time steps in $[r \cdot n, (r+1) \cdot n)$. We assume that the load vector $\boldsymbol{L}(r \cdot n) = \boldsymbol{\ell}$ at the beginning of round r is fixed. By recursive application of Equation (6), we get

$$\begin{aligned} &\mathbb{E}[\Phi(\boldsymbol{L}((r+1) \cdot n)) \mid \boldsymbol{L}(r \cdot n) = \boldsymbol{\ell}] \\ &\quad \leq \left(1 - \frac{1}{n}\right)^n \cdot \Phi(\boldsymbol{\ell}) + \frac{1}{2} \cdot \sum_{i=0}^{n-1} \left(1 - \frac{1}{n}\right)^i \\ &\quad \leq e^{-1} \cdot \Phi(\boldsymbol{\ell}) + \frac{n}{2} \cdot \left(1 - e^{-1}\right) \ , \end{aligned} \tag{7}$$

where we used the inequality $(1 - 1/n)^n \leq e^{-1}$. As long as $\Phi(\boldsymbol{\ell}) \geq n$, the expected potential after one round is therefore at most

$$\begin{aligned} &\mathbb{E}[\Phi(\boldsymbol{L}((r+1) \cdot n)) \mid \boldsymbol{L}(r \cdot n) = \boldsymbol{\ell}] \\ &\quad \leq \left(e^{-1} + \frac{n}{2\Phi(\boldsymbol{\ell})}\left(1 - e^{-1}\right)\right) \cdot \Phi(\boldsymbol{\ell}) \\ &\quad \leq \frac{1 + e^{-1}}{2} \cdot \Phi(\boldsymbol{\ell}) < \frac{3}{4} \cdot \Phi(\boldsymbol{\ell}) \ . \end{aligned} \tag{8}$$

Applying the Markov inequality (see Appendix A) now gives us, for a load vector $\boldsymbol{\ell}$ with $\Phi(\boldsymbol{\ell}) \geq n$,

$$\Pr\left[\Phi(\boldsymbol{L}((r+1) \cdot n)) \geq \frac{7}{8} \cdot \Phi(\boldsymbol{\ell}) \;\middle|\; \boldsymbol{L}(r \cdot n) = \boldsymbol{\ell}\right] \leq \frac{6}{7} \ . \tag{9}$$

We define for a round r the event that r is *successful* as $\Phi(\boldsymbol{L}((r+1) \cdot n)) \leq 7/8 \cdot \Phi(\boldsymbol{L}(r \cdot n)) \ \vee \ \Phi(\boldsymbol{L}(r \cdot n)) < n$ and use $\mathcal{E}_r$ to denote this event. Equation (9) implies $\Pr[\mathcal{E}_r] \geq 1/7$.

We now argue that after at most $\rho = \log_{8/7}(\Phi(\boldsymbol{L}(0))/n) + 1$ successful rounds the potential is smaller than n. Let r_ρ be the ρ-th successful round. There are two cases. If there exists a round $r \leq r_\rho$ for which $\Phi(\boldsymbol{L}(r \cdot n)) < n$, then $\Phi(\boldsymbol{L}(r_\rho)) < n$ is trivially true since the potential does never increase. Otherwise, by definition of a successful round, after ρ successful rounds we have

$$\begin{aligned} \Phi(\boldsymbol{L}(r_\rho \cdot n)) &\leq \left(\frac{7}{8}\right)^\rho \cdot \Phi(\boldsymbol{L}(0)) \\ &= \frac{7}{8} \cdot \frac{n}{\Phi(\boldsymbol{L}(0))} \cdot \Phi(\boldsymbol{L}(0)) < n \ . \end{aligned} \tag{10}$$

It remains to show that, w.h.p., during the first $O(\log n + \log \Delta)$ rounds at least ρ rounds are successful. Let the random variable X denote the number of successful rounds during the first $168(\ln n + \log \Delta)$ rounds. Since each round is successful with probability at least[3] $1/7$, the random variable X stochastically dominates the binomial random variable $Y \sim \mathrm{Bin}(168(\ln n + \log \Delta), 1/7)$ (written $X \succeq Y$). Applying Chernoff bounds (see Appendix A) to Y with its expected value $\mu = \mathbb{E}[Y] = 24(\ln n + \log \Delta)$ gives

$$\begin{aligned} \Pr[Y \leq \rho] &= \Pr\left[Y \leq \left(1 - \frac{\mu - \rho}{\mu}\right)\mu\right] \\ &\leq \exp\left(-\frac{(\mu - \rho)^2}{\mu^2} \cdot \frac{\mu}{2}\right) \\ &\overset{(12)}{\leq} \exp\left(-\frac{\mu}{8}\right) \leq n^{-3} \ , \end{aligned} \tag{11}$$

where we used the following inequality to bound $\mu - \rho$, holding for $\Delta \geq 1$:

$$\begin{aligned} \rho &= \log_{8/7}\left(\frac{\Phi(\boldsymbol{L}(0))}{n}\right) + 1 \\ &\leq \log_{8/7}\left(\frac{n \cdot \Delta^2}{n}\right) + 1 \\ &\leq \frac{2 \log \Delta}{\log 8/7} + 1 \\ &< 12 \log \Delta + 1 < \frac{\mu}{2} \ . \end{aligned} \tag{12}$$

Since $X \succeq Y$, Equation (11) implies that the probability of having fewer than ρ successful rounds during the first $7n \cdot \mu$ time steps is smaller than n^{-3}. Therefore, w.h.p.,

$$T_1 \leq 7n \cdot \mu = O(n \log n + n \log \Delta) \ . \qquad \square$$

B. Phase 2: Improving Individual Tokens

We now consider individual tokens. We start our analysis with Lemma 2, where we show that during any time step any token with normalized height larger than some constant reduces its height with probability $\Omega(1/n)$ by a constant factor. This is then used in Lemma 3 to argue that it takes at most $O(n \log n)$ time steps for all tokens to reach a constant normalized height.

For the sake of the analysis we now define which tokens are selected to be transferred when two nodes are balanced. Recall that according to the definition of the process tokens are indistinguishable and therefore arbitrary tokens may be selected.

Fix a time step t and assume that node u interacts with node v. In order to balance their loads, we need to move tokens from the node with larger load to the node with smaller load (say from u to v). To do so, we start with the token at maximal height and take every other token until we have selected required number of tokens. Then we place all tokens on node v in their original order. An example for this process is sketched in Figure 1b.

For the remainder, let $c \geq 10$ be a constant and recall that T_1 is the first time step of the second phase. The rule defined above allows us to show the following lemma.

Lemma 2. *Let $t \geq T_1$ and let b be a token with normalized height $\hat{h}_b(t) > 2c$. Then $\hat{h}_b(t+1) \leq 17/20 \cdot \hat{h}_b(t)$ with probability at least $1/n$.*

Proof. The idea of the proof is as follows. We first argue that at any time after the first phase fewer than half of the nodes have load larger than or equal to $\varnothing + c$. This is then used

[3] The lower bound holds independently for each round.

to derive a lower bound on the probability that a token of normalized height larger than $2c$ takes part in balancing with a node that has load at most $\varnothing + c$. Finally, we compute the new height of the token, which yields the lemma.

We now give the formal proof. Let $S(t) = \{ v \mid \ell_v(t) \geq \varnothing + c \}$ be the set of nodes which have load at least $\varnothing + c$ and suppose that $|S(t)| \geq n/2$. Then

$$\begin{aligned}\Phi(\boldsymbol{L}(t)) &= \sum_{i=1}^{n} (\ell_i(t) - \varnothing)^2 \geq \sum_{i \in S} (\ell_i(t) - \varnothing)^2 \\ &\geq \sum_{i \in S} c^2 \geq 100n/2 > n \ .\end{aligned} \tag{13}$$

However, the potential function does not increase over time and, thus, Lemma 1 implies that $\Phi(\boldsymbol{L}(t)) \leq n$ for any $t \geq T_1$. This is a contradiction and, therefore, $|S(t)| < n/2$.

We now proceed to lower bound the probability that b reduces its normalized height by a constant factor. Let i be the node on which token b is stored at time t. With probability $2/n$, node i is selected as one of the two nodes for balancing. Let furthermore j be the other node selected for balancing. Since $|S(t)| < n/2$, node j has load at most $\varnothing + c$ with probability at least $1/2$ (independent of i's selection). In that case, either $\left\lfloor \frac{\ell_i(t) - \ell_j(t)}{2} \right\rfloor$ or $\left\lceil \frac{\ell_i(t) - \ell_j(t)}{2} \right\rceil$ tokens are moved, depending on whether (i, j) or (j, i) are selected. Using that each other token is moved (see Figure 1b), carefully bounding the new height gives in both cases, whether b is transferred to node j or not, that the new height of token b becomes at most

$$\begin{aligned}h_b(t+1) &\leq \ell_j(t) + \left\lceil \frac{h_b(t) - \ell_j(t) + 1}{2} \right\rceil + 1 \\ &\leq \ell_j(t) + \frac{h_b(t) - \ell_j(t)}{2} + 2 \\ &= \frac{h_b(t) + \ell_j(t) + 4}{2} \ .\end{aligned} \tag{14}$$

We now bound the ratio between the new and the old normalized height of token b. For $\ell_j(t) \leq \lfloor \varnothing \rfloor + c$ and $h_b(t) \geq \lfloor \varnothing \rfloor + 2c$, this ratio is at most

$$\begin{aligned}\frac{\hat{\mathrm{h}}_b(t+1)}{\hat{\mathrm{h}}_b(t)} &= \frac{\frac{1}{2}(h_b(t) + \ell_j(t) + 4) - \lfloor \varnothing \rfloor}{h_b(t) - \lfloor \varnothing \rfloor} \\ &= \frac{1}{2} + \frac{1}{2} \cdot \frac{\ell_j(t) - \lfloor \varnothing \rfloor + 4}{h_b(t) - \lfloor \varnothing \rfloor} \\ &\leq \frac{1}{2} + \frac{c+4}{4c} \leq 0.85 \ ,\end{aligned} \tag{15}$$

where the last inequality holds since $c \geq 10$. Therefore, at any time $t \geq T_1$ and for any token b with $\hat{\mathrm{h}}_b(t) \geq 2c$, we have $\hat{\mathrm{h}}_b(t+1) \leq 0.85 \cdot \hat{\mathrm{h}}_b(t)$ with probability at least $1/n$. □

We are now ready to show the main lemma for Phase 2.

Lemma 3. *Let T_2 be the first time for which*

$$\max_{1 \leq i \leq n} \{ \ell_i(T_2) \} \leq \varnothing + 2c$$

and

$$\min_{1 \leq i \leq n} \{ \ell_i(T_2) \} \geq \varnothing - 2c \ .$$

With high probability, $T_2 = T_1 + \mathrm{O}(n \log n)$.

Proof. We first show the claim for the maximal load and then use a coupling argument to extend the analysis to the minimal load. For the maximal load, we consider a fixed token b and use Lemma 2 to define and bound the probability of a *successful* time step w.r.t. b. Then we show that this event occurs sufficiently often during the first $\mathrm{O}(n \log n)$ time steps such that b reaches normalized height at most $2c$ with high probability. Finally, we show the claim by a union bound over all tokens of normalized height larger than $2c$.

Let b be an arbitrary but fixed token with $\hat{\mathrm{h}}_b(t) \geq 2c$. We call a time step t *successful* if $\hat{\mathrm{h}}_b(t+1) \leq 17/20 \cdot \hat{\mathrm{h}}_b(t) \vee \hat{\mathrm{h}}_b(t) \leq 2c$. From Lemma 2 we get that time step t is successful with probability at least $1/n$. Note that while the behavior of two different tokens may be highly correlated, for one fixed token the lower bounds hold independently for any time step in the second phase. This allows us to leverage stochastic dominance of a binomial distribution as follows: Let the random variable $X_b(\tau)$ denote the number of successful time steps during the first τ time steps in the second phase. Since each time step is successful with probability at least $1/n$, the random variable $X_b(\tau)$ stochastically dominates the binomial random variable $Y_b(\tau) \sim \mathrm{Bin}(\tau, 1/n)$. Applying Chernoff bounds (see Appendix A) to $Y_b(\tau)$ with $\tau = 12n \log n$ gives

$$\begin{aligned}&\Pr\left[Y_b(12n \log n) \leq \left(1 - \frac{3}{4}\right) \mathbb{E}[Y_b(12n \log n)] \right] \\ &\quad \leq \exp\left(-\frac{1}{2} \cdot \frac{9}{16} \cdot 12 \log n \right) \leq n^{-3} \ .\end{aligned} \tag{16}$$

With the above mentioned stochastic dominance $X_b(\tau) \succeq Y_b(\tau)$, we get that $X_b(12n \log n) \leq 3 \log n$ with probability at most n^{-3}. It remains to show that the normalized height of b after $3 \log n$ successful time steps is at most $2c$. Observe that $\hat{\mathrm{h}}_b(T_1) \leq \sqrt{n}$, since otherwise $\Phi(\boldsymbol{L}(T_1)) \geq n$. Therefore, after at most $3 \log n$ successful time steps in the second phase, the normalized height of b is at most[4]

$$\hat{\mathrm{h}}_b(T_1 + 12n \log n) \leq \max \{ \sqrt{n} \cdot \left(\frac{17}{20}\right)^{3 \log n}, 2c \} \leq 2c \ . \tag{17}$$

We now use the union bound on the above analysis over all tokens as follows. From the bound on the potential function in Lemma 1 we obtain that after the first phase at most n tokens remain above the average, since otherwise the potential would be larger than n. Observing that the height of a token never increases and taking the union bound over all tokens of normalized height above $2c$ gives us that all tokens have remaining height at most $2c$ after at most $12n \log n$ interactions with probability $1 - 1/n^{-2}$.

We now argue an analogous bound for the minimal load. Let $\boldsymbol{\ell} \in \mathbb{Z}^n$ be the initial load vector of the load balancing process $\boldsymbol{L}(0) = \boldsymbol{\ell}, \boldsymbol{L}(1), \boldsymbol{L}(2), \ldots$ and let $-\boldsymbol{\ell}$ be the initial load vector of the load balancing process $\boldsymbol{L}'(0) = -\boldsymbol{\ell}, \boldsymbol{L}'(1), \boldsymbol{L}'(2), \ldots$. We can couple the processes such that whenever a pair of nodes (u, v) is chosen in $\boldsymbol{L}(t)$, the pair of nodes (v, u) is chosen in

[4] The maximum in Equation (17) covers the fact that the analysis does not extend to $\hat{\mathrm{h}}_b(t) < 2c$.

$\boldsymbol{L}'(t)$. This coupling ensures (determinstically) that $\ell_i(t) = -\ell'_i(t)$ and, thus, implies $\Pr[\ell_i(t) = x] = \Pr[\ell'_i(t) = -x]$. By applying the upper bound on the maximal load to $\boldsymbol{L}'(T_1 + 12n \log n)$, we get a lower bound on the minimal load in $\boldsymbol{L}(T_1 + 12n \log n)$. We therefore get that $T_2 \leq T_1 + \mathrm{O}(n \log n)$, which concludes the proof. □

C. Phase 3: Fine Tuning

For the sake of the analysis of the third phase, we use the following rule to select tokens to transfer when balancing two nodes. We again assume that nodes operate like stacks, with the following additional rule: both nodes shuffle their tokens of normalized height in $\{ 2, 3, \ldots, 2c \}$ (if they exist) before balancing the loads. This rule allows us to show the following lemma, our main result.

Lemma 4. *Let T_3 be the first time for which*

$$\max_{1 \leq i \leq n} \{ \ell_i(T_3) \} \leq \lfloor \varnothing \rceil + 1$$

and

$$\min_{1 \leq i \leq n} \{ \ell_i(T_3) \} \geq \lfloor \varnothing \rceil - 1 \ .$$

With high probability, $T_3 = T_2 + \mathrm{O}(n \log n)$.

Proof. We again start by analyzing the maximal load. We first show that at any time step after the second phase at least a constant fraction of nodes has load at most $\lfloor \varnothing \rceil$. Then we consider an arbitrary but fixed token b with $\hat{\mathrm{h}}_b(t) > 1$ at time t and show that with probability $\Omega(1/n)$ we have $\hat{\mathrm{h}}_b(t+1) \leq 1$. This is used to show that, w.h.p., $\hat{\mathrm{h}}_b(\tau) \leq 1$ for $\tau = \mathrm{O}(n \log n)$. The claim then follows from a union bound over all tokens above normalized height 1.

Fix a time step $t \geq T_2$ and let γ be the fraction of nodes that have load at most $\lfloor \varnothing \rceil$ at time t. We use the definition of the rounded average load and Lemma 3 to compute

$$\begin{aligned} & n \cdot (\lfloor \varnothing \rceil + 0.5) \\ & \quad \geq n \cdot \varnothing = \sum_{1 \leq i \leq n} \ell_i(t) = \sum_{\ell_i(t) > \lfloor \varnothing \rceil} \ell_i(t) + \sum_{\ell_i(t) \leq \lfloor \varnothing \rceil} \ell_i(t) \\ & \quad \geq \sum_{\ell_i(t) > \lfloor \varnothing \rceil} (\lfloor \varnothing \rceil + 1) + \sum_{\ell_i(t) \leq \lfloor \varnothing \rceil} (\lfloor \varnothing \rceil - 2c) \\ & \quad \geq n \cdot (1 - \gamma) \cdot (\lfloor \varnothing \rceil + 1) + n \cdot \gamma \cdot (\lfloor \varnothing \rceil - 2c) \ . \end{aligned} \tag{18}$$

Therefore, solving for γ gives us that γ is a constant depending on c with $\gamma \geq 1/(2 + 4c)$.

Similar to the analysis of the second phase, we now consider an arbitrary but fixed token b. Fix a time step $t \geq T_2$ and a token b with $\hat{\mathrm{h}}_b(t) > 1$. Let i be the node on which b resides before time step t. We have the following events.

a) Node i is selected for balancing: in any time step, i is selected with probability $2/n$.
b) Token b becomes the top-most token: all tokens b' on node i of normalized height $\hat{\mathrm{h}}_{b'}(t) > 1$ are shuffled. Since there are at most $2c$ such tokens after the second phase, b becomes the top-most token with probability $\geq 1/(2c)$.
c) The other node has load at most $\lfloor \varnothing \rceil$: since the fraction of such nodes is least γ, such a node is selected as the balancing partner with probability at least γ.

We say b is *successful* in time step t if all three of these events occur. Observe that in this case $\hat{\mathrm{h}}_b(t+1) \leq 1$. Let $p_b(t)$ be the probability of a successful time step. Combining above probabilities, we get $p_b(t) \geq 2/n \cdot 1/(2c) \cdot \gamma = \Omega(1/n)$.

We now consider $\mathrm{O}(n \log n)$ time steps after the second phase. Token b is not successful at least once during these time steps with probability

$$\prod_{t=1}^{\mathrm{O}(n \log n)} (1 - p_b(t)) \leq \left(1 - \Omega\left(\frac{1}{n}\right)\right)^{\mathrm{O}(n \log n)} \leq n^{-\Omega(1)} \ . \tag{19}$$

That is, for a suitable choice of constants, b reaches height 1 after at most $\mathrm{O}(n \log n)$ time steps with probability $1 - 1/n^3$. The upper bound on the load now follows from a union bound, since at most $2c \cdot n$ tokens have normalized height above 1 after the second phase. For the lower bound on the load, precisely the same argument as in the proof of Lemma 3 can be used. □

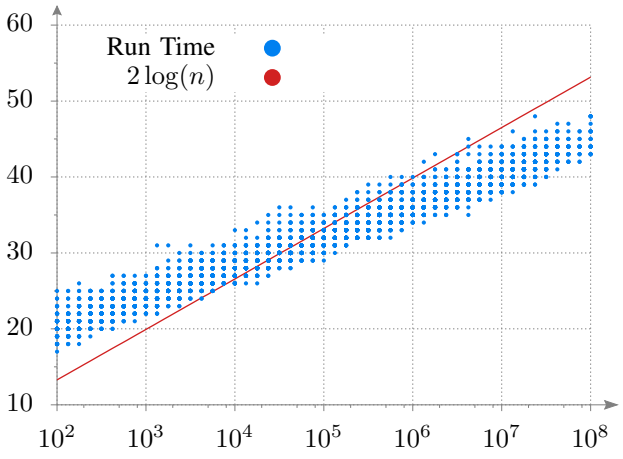
The proof of Theorem 1 now follows from a union bound over the results from Lemma 1 for the first phase, Lemma 3 for the second phase, and Lemma 4 for the third phase.

III. Empirical Analysis

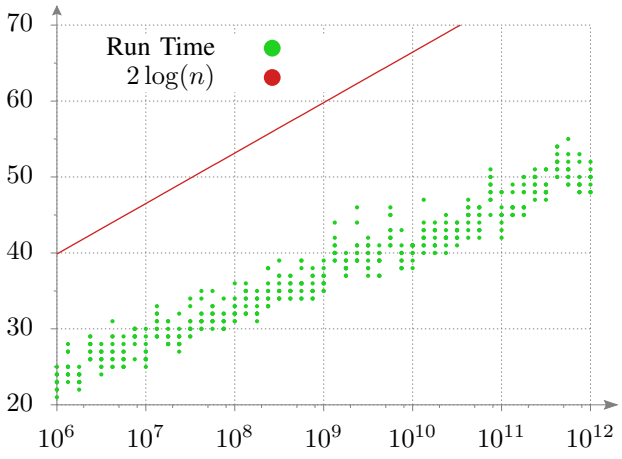
In this section we present simulation results to support our theoretical findings. We implemented our simulation software to run on a shared memory machine and simulate the distributed system. The simulation is written in the C++ programming language and consists of roughly 100 lines of code. It allows to initially set n, the number of nodes, and m, the number of tokens. Assuming the worst case, all load items are initially placed on a single node such that $\Delta = m$. The load balancing procedure is then simulated until the remaining discrepancy is at most 2 (or 1 in the case of Figure 3). To allow a more efficient simulation, the implementation checks only once per n interactions whether the process has already converged. Consistently, the simulation returns only the round – the number of interactions divided by n – as its run time.

The two plots in Figure 2 both show the run time (in rounds of n interactions each) of the simulation. In both plots, the required number of rounds until the remaining discrepancy is at most 2 is shown on the y-axis. In Figure 2a, we simulated the load balancing procedure for varying numbers of nodes n on the x-axis, while the load was set such that $\varnothing = 1000$. In Figure 2b, the number of nodes was set to a fixed number of $n = 10^6$, while the load runs from $m = 10^6$ to $m = 10^{12}$. For each data point on the x-axis, more than 10 independent simulation runs have been executed. The red lines in Figure 2a show the functions $f(n) = 2 \log n$ and $g(n) = 2 \log \Delta$. Altogether, the simulations do not only confirm our theoretical results but show that the constants of the process are small in practice. In particular, the measured run times have been well below $3 \cdot n(\log n + \log \Delta)$ interactions.

Additionally, in Figure 3 we analyzed the run time until the system has *fully converged* and all nodes have load either $\lfloor \varnothing \rfloor$ or $\lceil \varnothing \rceil$. In this case, the remaining discrepancy is either 0 if $\varnothing$ is an integer or 1 otherwise, and this is the best that can be



(a) Number of Nodes n



(b) Initial Discrepancy Δ

Fig. 2: Empirical analysis. Both plots show the run time in rounds (y-axis) of n interactions until the system has remaining discrepancy at most 2. In the left plot, the system was initialized such that $\varnothing = 1000$ for varying n (x-axis). In the right plot, n was set to 10^6 and the run time was measured for varying initial discrepancy Δ (x-axis). In both plots, the entire load was initially assigned to a single node.

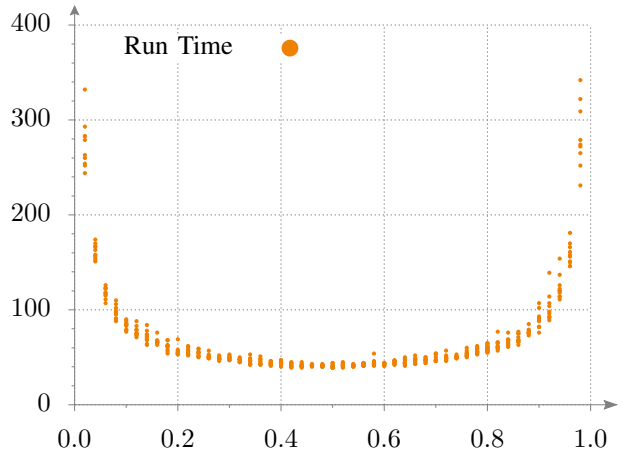


Fig. 3: Empirical analysis. The plot shows the run time to *full convergence* for varying fractional parts of $\varnothing$.

achieved in the discrete setting. In Figure 3, we simulated the load balancing process in a system of 10^6 nodes. On the y-axis, we plotted the run time until all nodes have load either $\lfloor\varnothing\rfloor$ or $\lceil\varnothing\rceil$. The average load $\varnothing$ was selected from the interval $[1000, 1001]$ and the fractional part $\varnothing - \lfloor\varnothing\rfloor$ of $\varnothing$ is shown on the x-axis. As before, the entire load was initially placed on one single node such that $\Delta = m$. The plots indicate that for a large range of parameters m and n the process fully converges within $4 \cdot n(\log n + \log \Delta)$ interactions.

References

[1] H. Ackermann, S. Fischer, M. Hoefer, and M. Schöngens. "Distributed algorithms for QoS load balancing". In: *Distributed Computing* 23.5-6 (2011), pp. 321–330. DOI: 10.1007/s00446-010-0125-1.

[2] H. Akbari, P. Berenbrink, and T. Sauerwald. "A simple approach for adapting continuous load balancing processes to discrete settings". In: *Distributed Computing* 29.2 (2016), pp. 143–161. DOI: 10.1007/s00446-016-0266-y.

[3] P. Berenbrink, C. Cooper, T. Friedetzky, T. Friedrich, and T. Sauerwald. "Randomized diffusion for indivisible loads". In: *J. Comput. Syst. Sci.* 81.1 (2015), pp. 159–185. DOI: 10.1016/j.jcss.2014.04.027.

[4] P. Berenbrink, T. Friedetzky, and Z. Hu. "A new analytical method for parallel, diffusion-type load balancing". In: *J. Parallel Distrib. Comput.* 69.1 (2009), pp. 54–61. DOI: 10.1016/j.jpdc.2008.05.005.

[5] P. Berenbrink, P. Kling, C. Liaw, and A. Mehrabian. "Tight Load Balancing via Randomized Local Search". In: *Proceedings of the 31st International Parallel & Distributed Processing Symposium (IPDPS)*. 2017, pp. 192–201. DOI: 10.1109/IPDPS.2017.52.

[6] R. Diekmann, A. Frommer, and B. Monien. "Efficient schemes for nearest neighbor load balancing". In: *Parallel Computing* 25.7 (1999), pp. 789–812. DOI: 10.1016/S0167-8191(99)00018-6.

[7] S. Fischer, H. Räcke, and B. Vöcking. "Fast Convergence to Wardrop Equilibria by Adaptive Sampling Methods". In: *SIAM J. Comput.* 39.8 (2010), pp. 3700–3735. DOI: 10.1137/090746720.

[8] T. Friedrich and T. Sauerwald. "Near-perfect load balancing by randomized rounding". In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. 2009, pp. 121–130. DOI: 10.1145/1536414.1536433.

[9] B. Ghosh and S. Muthukrishnan. "Dynamic Load Balancing by Random Matchings". In: *J. Comput. Syst. Sci.* 53.3 (1996), pp. 357–370. DOI: 10.1006/jcss.1996.0075.

[10] P. W. Goldberg. "Bounds for the Convergence Rate of Randomized Local Search in a Multiplayer Load-balancing Game". In: *Proceedings of the 23rd Annual Symposium on Principles of Distributed Computing (PODC)*. 2004, pp. 131–140. DOI: 10.1145/1011767.1011787.

[11] M. Hoefer and T. Sauerwald. "Threshold Load Balancing in Networks". In: *CoRR* abs/1306.1402 (2013). arXiv: 1306.1402.

[12] D. Kempe, A. Dobra, and J. Gehrke. "Gossip-Based Computation of Aggregate Information". In: *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*. 2003, pp. 482–491. DOI: 10.1109/SFCS.2003.1238221.

[13] M. Mitzenmacher and E. Upfal. *Probability and Computing – Randomized Algorithms and Probabilistic Analysis*. 2005.

[14] Y. Mocquard, E. Anceaume, and B. Sericola. "Optimal Proportion Computation with Population Protocols". In: *Proceedings of the 15th International Symposium on Network Computing and Applications (NCA)*. 2016, pp. 216–223. DOI: 10.1109/NCA.2016.7778621.

[15] S. Muthukrishnan, B. Ghosh, and M. H. Schultz. "First- and Second-Order Diffusive Methods for Rapid, Coarse, Distributed Load Balancing". In: *Theory Comput. Syst.* 31.4 (1998), pp. 331–354. DOI: 10.1007/s002240000092.

[16] Y. Rabani, A. Sinclair, and R. Wanka. "Local Divergence of Markov Chains and the Analysis of Iterative Load Balancing Schemes". In: *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS)*. 1998, pp. 694–705. DOI: 10.1109/SFCS.1998.743520.

[17] T. Sauerwald and H. Sun. "Tight Bounds for Randomized Load Balancing on Arbitrary Network Topologies". In: *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS)*. 2012, pp. 341–350. DOI: 10.1109/FOCS.2012.86.

APPENDIX

A. Tail Bounds

For completeness, we formally state in this appendix the probabilistic tools that we use in our analysis. The following theorem is the conditional version of Markov's inequality, which is based on Theorem 3.1 from [13].

Theorem 2 (Markov's Inequality)**.** *Let X be a random variable that assumes only nonnegative values defined for a probability space Ω. Then, for all $a > 0$ and events A,*

$$\Pr[X \geq a \mid A] \leq \frac{\mathbb{E}[X \mid A]}{a} \ .$$

The following versions of Chernoff bounds are a combination of Theorem 4.4 and Theorem 4.5 from [13].

Theorem 3 (Chernoff Bounds)**.** *Let $X_1, \ldots, X_n$ be independent Poisson trials such that $\Pr[X_i] = p_i$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X]$. Then for $0 < \delta < 1$*

$$\Pr[X \geq (1+\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{3}\right)$$

and

$$\Pr[X \leq (1-\delta)\mu] \leq \exp\left(-\frac{\delta^2\mu}{2}\right) \ .$$

B. Full Equations

In the following, we show Equation (5).

Lemma 5. *Let $\Phi(\boldsymbol{\ell})$ be the potential function of a load vector $\boldsymbol{\ell}$, defined as*

$$\Phi(\boldsymbol{\ell}) = \sum_{i=1}^{n} (\ell_i - \varnothing)^2 \ .$$

Then

$$\sum_{i=1}^{n} \sum_{j=i+1}^{n} (\ell_i - \ell_j)^2 = n \cdot \Phi(\boldsymbol{\ell}) \ .$$

Proof. Since $(\ell_i - \ell_j)^2 = 0$ for $i = j$ we get by symmetry

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{j=i+1}^{n} (\ell_i - \ell_j)^2 &= \frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} (\ell_i - \ell_j)^2 \\
&= \frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} (\ell_i^2 - 2\ell_i\ell_j + \ell_j^2) \\
&= \frac{1}{2} \cdot 2 \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \ell_i^2 - \frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} 2\ell_i\ell_j \\
&= n \cdot \sum_{i=1}^{n} \ell_i^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} \ell_i\ell_j \\
&= n \cdot \sum_{i=1}^{n} \ell_i^2 - n \cdot \sum_{i=1}^{n} \left(\ell_i \cdot \sum_{j=1}^{n} \frac{\ell_j}{n} \right) \ .
\end{aligned}$$

We now use the definition $\varnothing = \sum_{i=1}^{n} \frac{\ell_i}{n}$ to get

$$\begin{aligned}
\sum_{i=1}^{n} \sum_{j=i+1}^{n} (\ell_i - \ell_j)^2 &= n \cdot \left(\sum_{i=1}^{n} \ell_i^2 - \sum_{i=1}^{n} \ell_i\varnothing \right) \\
&= n \cdot \left(\sum_{i=1}^{n} \ell_i^2 - 2 \cdot \sum_{i=1}^{n} \ell_i\varnothing + \sum_{i=1}^{n} \ell_i\varnothing \right) \\
&= n \cdot \left(\sum_{i=1}^{n} \ell_i^2 - 2 \cdot \sum_{i=1}^{n} \ell_i\varnothing + n \cdot \varnothing^2 \right) \\
&= n \cdot \left(\sum_{i=1}^{n} \ell_i^2 - 2 \cdot \sum_{i=1}^{n} \ell_i\varnothing + \sum_{i=1}^{n} \varnothing^2 \right) \\
&= n \cdot \sum_{i=1}^{n} (\ell_i^2 - 2\ell_i\varnothing + \varnothing^2) \\
&= n \cdot \sum_{i=1}^{n} (\ell_i - \varnothing)^2 = n \cdot \Phi(\boldsymbol{\ell}) \ .
\end{aligned}$$

□