Strongly Adaptive Token Distribution

F. Meyer auf der Heide, B. Oesterdiekhoff, and R. Wanka

Abstract. The token distribution (TD) problem, an abstract static variant of load balancing, is defined as follows: let $M$ be a (parallel processor) network with set $\mathcal{P}$ of processors. Initially, each processor $P \in \mathcal{P}$ has a certain amount $l(P)$ of tokens. The goal of a TD algorithm, run on $M$, is to distribute the tokens evenly among the processors. In this paper we introduce the notion of strongly adaptive TD algorithms, i.e., algorithms whose running times come close to the best possible runtime, the off-line complexity of the TD problem, for each individual initial token distribution $l$. Until now, only weakly adaptive algorithms have been considered, where the running time is measured in terms of the maximum initial load $\max\{|l(P)| P \in \mathcal{P}\}$.

We design an almost optimal, strongly adaptive algorithm on mesh-connected networks of arbitrary dimension extended by a single 1-bit bus. This result shows that an on-line TD algorithm can come close to the optimal (off-line) bound for each individual initial load. Furthermore, we exactly characterize the off-line complexity of arbitrary initial token distributions on arbitrary networks. As an intermediate result, we design almost optimal weakly adaptive algorithms for TD on mesh-connected networks of arbitrary dimension.

Key Words. Parallel algorithms, Token distribution.

1. Introduction.

1.1. Computation Model. The underlying parallel computation model is the (parallel processor) network. Such a network consists of a set of $p$ processors, $\mathcal{P} = \{P_1, \ldots, P_p\}$ pairs of which are connected via bidirectional links forming a communication graph $M = (\mathcal{P}, E)$ with $E$ denoting the set of links. We identify the network with its communication graph. The network is assumed to be synchronized; in a computation step, each processor can do a constant amount of internal computation and can send a message to a neighboring processor. We demand that each processor can receive at most one message per step.

1.2. The Token Distribution Problem. Load balancing is one of the basic tasks to be performed in order to achieve efficient execution of parallel programs on a network: if a processor is overloaded with work during a computation, it tries to reduce its load by shifting part of it to less busy processors. The aim is to keep the load balanced among the processors.

Token distribution (TD) is an abstraction of a static variant of load balancing. Initially each processor $P$ has a number $l(P)$ of tokens, i.e., the initial load is given by a function $l: \mathcal{P} \rightarrow \mathbb{N}$. We refer to $N = \sum_{P \in \mathcal{P}} l(P)$ as the total load, to $k = \max\{|l(P)| P \in \mathcal{P}\}$ as the maximum load, and to $N/p$ as the average load.

1 This research was partially supported by DFG-Forschergruppe "Effiziente Nutzung massiv paralleler Systeme, Teilprojekt 4," by the ESPRIT Basic Research Action No. 7141 (ALCOM II), and by the Volkswagenstiftung. A preliminary version was presented at the 20th ICALP, 1993, see [9].

2 Fachbereich Mathematik-Informatik and Heinz-Nixdorf-Institut, Universität-GH Paderborn, D-33095 Paderborn, Germany. \{fmadh, brigitte, wanka\}@uni-paderborn.de.

Received November 9, 1993; revised May 11, 1994, and August 21, 1994. Communicated by K. Mehlhorn.
The goal of token distribution is to distribute the tokens given by the initial load so that the final load of each processor is close to the average load. More specifically, a TD algorithm is $\delta$-exact if finally no processor holds more than $\lceil N/p \rceil + \delta$ tokens. Furthermore, we denote the maximum load difference of the processors with discrepancy.

1.3. Complexity Measures of Token Distribution. In order to be able to measure the performance of a TD algorithm, we first introduce a quantity which is a lower bound on the running time of any TD algorithm on $M$ with initial load $l$. For this purpose, we consider off-line algorithms for token distribution. In this case, with $M, l$ given, we allow an arbitrarily complex preprocessing that can be executed without being added to the complexity and that produces a protocol for each processor telling, for each time $t$, whether and, if yes, where to send a token. We assume that a processor can send and receive one token per time step. The off-line complexity of the TD problem $(M, l)$, the TD problem on $M$ with initial load $l$, is $T^{\text{off}}(M; l, \delta)$, the running time of a fastest $\delta$-exact off-line TD algorithm.

On-line algorithms are designed for a fixed network $M$, but do not allow any free preprocessing given an initial load $l$. Thus, a TD algorithm executes distribution steps, where each processor can send and receive one token, and computation steps, where computation and communication can take place (e.g., to gather information about the current load distribution). No tokens are moved in such a step. The on-line complexity of the TD problem $(M, l)$ is $T(M; l, \delta)$, the running time of a fastest $\delta$-exact on-line TD algorithm on $M$ started with initial load $l$.

All papers mentioned below consider the following adaptive complexity measure for TD on $M$:

$$T_{\text{ad}}(M; N, k, \delta) = \max\{T(M; l, \delta) | l \text{ has maximum load } \leq k \text{ and total load } \leq N\}.$$

In this paper we consider an even stronger version of adaptivity: we want to design on-line TD algorithms which come close to the performance of off-line algorithms of each individual initial load function, i.e., we design strongly adaptive TD algorithms, showing that their performance comes close to the lower bound, i.e., the off-line complexity. The strongly adaptive complexity is the running time of a fastest $\delta$-exact on-line TD algorithm depending on the off-line complexity $T^{\text{off}}(M; l, \delta)$, the network $M$, and the maximum load $k$ of the individual load $l$. We assume that the total load $N$ is initially known to all processors.

1.4. Known Results About Token Distribution. The token distribution problem was introduced by Peleg and Upfal [10]. For arbitrary networks $M$ with $p$ processors and total load $N = p$, the same authors show in [11] that, for all $k \geq 2$,

$$T_{\text{ad}}(M; p, k, 0) = O(p) \quad \text{and} \quad T_{\text{ad}}(M; p, k, 0) = \Omega(k + \text{diam}(M)),$$

with $\text{diam}(M)$ denoting the diameter of $M$. 