Modern Parallel Algorithms

Artur Czumaj
University of Warwick, Coventry, UK

Abstract

Recent advances in the design of efficient parallel algorithms have been largely focusing on the nowadays classical model of parallel computing called *Massive Parallel Computation (MPC)*, which follows the framework of MapReduce systems. In this talk we will survey recent advances in the design of algorithms for graph problems for the MPC model and will mention some interesting open questions in this area.

2012 ACM Subject Classification Theory of computation → Massively parallel algorithms; Theory of computation → Distributed algorithms; Mathematics of computing → Graph algorithms

Keywords and phrases Distributed computing, parallel computing

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.3

Category Invited Talk

Funding *Artur Czumaj*: Research supported in part by the Centre for Discrete Mathematics and its Applications, by EPSRC award EP/V01305X/1, by a Weizmann-UK Making Connections Grant, by an IBM Award, and by the Simons Foundation Award No. 663281 granted to the Institute of Mathematics of the Polish Academy of Sciences for the years 2021–2023.

1 Overview

Many modern computer applications require performing computations on massive amounts of data. Traditional models of computation, such as the RAM model or shared-memory parallel systems, are often inadequate for such computations, as the input do not fit into the available memory of even most advanced modern systems. The restrictions imposed by the limited memory in the available architectures and the requirement of fast processing of data has naturally led to the development of new models of parallel and distributed computation that are more suitable for processing massive amounts of data. On the basis of the successes of such massively parallel computation frameworks, such as MapReduce, Hadoop, Dryad, or Spark, Karloff, Suri, and Vassilvitskii (SODA 2010) introduced the *Massive Parallel Computing (MPC) model* that provides a clean abstraction of these frameworks and captures the modern needs of computation at a massive scale. After some later refinements, the MPC model has became the standard theoretical model of algorithmic study. At a very high-level, an MPC system consists of a collection of machines that can communicate with each other through indirect communication channels. The computation proceeds in synchronous rounds, where at each round the machines receive messages from other machines, perform arbitrarily complex local computations, and finally send appropriate messages to other machines so that the next round can start. The crucial factors in the analysis of algorithms in the MPC model are the number of rounds and the capacity of individual machines.

In the MPC model, there are $\mathfrak{m}$ machines and each of them has $\mathfrak{s}$ words of local space at its disposal. Initially, each machine receives its share of the input. For example, in the context of graph problems where the input is a collection V of nodes and E of edges, the input is arbitrarily distributed among the machines (and so $\mathfrak{s} \cdot \mathfrak{m} \geq |V| + |E|$). The computation proceeds in synchronous *rounds* in which each machine processes its local data and performs an arbitrary complex local computation on its data. At the end of each round, machines exchange messages. Each message is sent only to a single machine specified by the machine

48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023).
Editors: Jérôme Leroux, Sylvain Lombardy, and David Peleg; Article No. 3; pp. 3:1–3:2
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

that is sending the message. All messages sent and received by each machine in each round, as well as the output have to fit into the machine's local space $\mathfrak{s}$ (in particular, in a single round, any machine can send at most $\mathfrak{s}$ messages and be the recipient of at most $\mathfrak{s}$ messages).

It has been quickly observed that the central parameter of the MPC is its local space $\mathfrak{s}$. While originally the main research has been frequently focused on the case when $\mathfrak{s}$ is almost as large as the input size, most recent study has been concentrated on the *low-space regime* when $\mathfrak{s} = N^{\phi}$ for some $\phi \in (0, 1)$, often ϕ being arbitrarily small.

The talk will survey this topic, focusing on graph problems for the low-space regime. We will discuss recent advances in the design of algorithms for graph problems for the MPC model for fundamental problems like connectivity and matching. We will also study the relation between the MPC model and some other fundamental models of parallel and distributing computations, including the classical PRAM model and the distributed LOCAL and Congested Clique models. We will also list some interesting open questions in this area.