Scheduling Data Gathering in 2-Level Tree Networks

Joanna Berlińska *

1 Introduction

Gathering data from remote processors is an important step of many contemporary applications. The data may be obtained as a result of computations or by sensing the environment. They have to be collected together for analysis, processing and storing. Scheduling algorithms for data gathering networks were proposed, e.g., in [1, 2, 5, 7].

In this work we analyze data gathering in a 2-level tree network. The leaf nodes of the network transfer data to intermediate nodes, which preprocess and merge them, constructing new, intermediate datasets. These datasets have to be sent to a single base station. The base station processes them and stores final results. Our goal is to schedule communication and computations in the network so that the whole data gathering process takes the shortest possible time.

2 Problem formulation

The data gathering network consists of a single base station P_0, n intermediate nodes $P_1, \ldots, P_n$ and m leaf nodes P_{jk}. An intermediate node P_j gathers data from leaf nodes P_{jk} for $k = 1, \ldots, m_j$, where $m_1 + \ldots + m_n = m$. A subnetwork consisting of node P_j and all nodes P_{jk} for given j will be denoted by N_j. It is assumed that all network nodes have identical communication and computation capabilities, described by communication rate (inverse of speed) C and computation rate A.

At time $t = 0$ each node P_{jk} holds dataset D_{jk} of size α_{jk}, which should be sent to P_j (in time $C\alpha_{jk}$). At most one leaf node can communicate with intermediate node P_j at a time. After the transfer of dataset D_{jk} is completed, this dataset can be preprocessed by P_j, in time $A\alpha_{jk}$. The results produced by P_j for all datasets D_{jk} are concatenated and constitute a new dataset D_j of size $\alpha_j = \gamma \sum_{k=1}^{m_j} \alpha_{jk}$, where γ is a parameter describing the relationship between the sizes of initial and intermediate data. Node P_j needs time $C\alpha_j$ to transfer this dataset to the base station, which processes it in time $A\alpha_j$. At most one intermediate node can communicate with the base station at a time. It is assumed that each dataset can be sent in many separate pieces, i.e. communication preemptions are allowed. The scheduling problem is to minimize the total data gathering time T.

*Joanna.Berlinska@amu.edu.pl. Faculty of Mathematics and Computer Science, Adam Mickiewicz University in Poznań, Umultowska 87, 61-614 Poznań, Poland.

3 Results

Let us first note that solving our scheduling problem can be divided in two steps. In the first step a schedule for data gathering in each subnetwork N_j will be constructed separately. As communication in a subnetwork is sequential, the analyzed problem consists in makespan minimization in a 2-machine flow shop, where the communication network is the first machine, and node P_j is the second machine. The two operations of job k ($k = 1, \ldots, m_j$) are sending and preprocessing dataset D_{jk}. Therefore, the problem can be solved in $O(m_j \log m_j)$ time using Johnson's algorithm [6]. The minimum makespan computed for subnetwork N_j will be denoted by r_j. Thus, intermediate node P_j is ready to start transferring data to the base station at time r_j. The second step is constructing a schedule for sending and processing the intermediate results. As the communication between intermediate nodes and the base station is sequential, we have to solve a special case of another flow shop scheduling problem, $F2|r_j, pmtn|C_{max}$, which is known to be strongly NP-hard in general [4]. Let us remind that in our problem the execution times of two operations of job j are equal to $p_{1j} = C\alpha_j$ and $p_{2j} = A\alpha_j$, i.e. we consider a proportionate flow shop with different machine speeds.

We first show that if $\alpha_j = \alpha$ for $j = 1, \ldots, n$, i.e. all subnetworks N_j deliver the same amounts of data, then the shortest schedule can be computed in $O(n \log n)$ time. Namely, every time the communication network becomes idle or a new dataset is released, an available dataset with the shortest remaining transfer time should be chosen for sending. Note that $\alpha_j = \alpha$ does not mean that release times r_j have to be equal, since the subnetworks may contain different numbers of leaf nodes holding datasets of different sizes α_{jk} which sum up to α/γ.

For the general case with different α_j we analyze the following communication scheduling algorithm.

1. Let $J = \{1, 2, \ldots, n\}$ and $t = \min_{j \in J}\{r_j\}$.

2. Find the set of available jobs $A = \{j | j \in J, r_j \leq t\}$ and $t' = \min\{r_j | r_j > t, j \in J\}$. Choose a job j in A according to Johnson's rule [6].

3. Let $L = \min\{p_{1j}, t' - t\}$. If $p_{1j} \leq L$, schedule the transfer of dataset D_j in interval $[t, t+p_{1j})$ and set $J = J \setminus \{j\}$. In the opposite case schedule the transfer of dataset D_j in interval $[t, t + L)$ and set $p_{1j} = p_{1j} - L$.

4. If $J \neq \emptyset$, set $t = t + L$ and go to step 2.

Thus, every time the communication network becomes idle or a new dataset is released, the dataset to be transferred is chosen according to Johnson's rule. The datasets are processed by the base station in the order in which they are received. We show that

- If $C \leq A$, the above algorithm computes the shortest possible schedule, since our problem boils down to a special case of $F2|r_j, 1\text{-}min, pmtn|C_{max}$ [3].

- If $C > A$, the algorithm does not always produce the optimum solution.

The case with $C > A$ is further analyzed in order to identify additional conditions under which the considered algorithm is guaranteed to deliver optimum results.

4 Conclusions and future work

We showed that the analyzed data gathering scheduling problem can be solved in polynomial time if $C \leq A$ or $\alpha_j = \alpha$ for $j = 1, \dots, n$. However, the complexity status of the case with $C > A$ and different α_j remains open and should be investigated in the future. A promising direction in constructing (exact or approximation) algorithms for this problem may be analyzing the relationship between intermediate dataset sizes α_j and their release times r_j.

References

[1] J. Berlińska, Communication scheduling in data gathering networks with limited memory, Applied Mathematics and Computation 235 (2014) 530-537.

[2] J. Berlińska, Scheduling for data gathering networks with data compression, European Journal of Operational Research 246 (2015) 744-749.

[3] J. Cheng, G. Steiner, P. Stephenson, A computational study with a new algorithm for the three-machine permutation flow-shop problem with release times, European Journal of Operational Research 130 (2001) 559-575.

[4] Y. Cho, S. Sahni, Preemptive Scheduling of Independent Jobs with Release and Due Times on Open, Flow and Job Shops, Operations Research 29 (1981) 511-522.

[5] K. Choi, T.G. Robertazzi, Divisible Load Scheduling in Wireless Sensor Networks with Information Utility, IEEE International Performance Computing and Communications Conference 2008: IPCCC 2008, 9-17.

[6] S.M. Johnson, Optimal two- and three-stage production schedules with setup times included, Naval Research Logistics Quarterly 1 (1954) 61-68.

[7] M. Moges, T.G. Robertazzi, Wireless Sensor Networks: Scheduling for Measurement and Data Reporting, IEEE Transactions on Aerospace and Electronic Systems 42 (2006) 327-340.