Data Distribution and Scheduling for Distributed Analytics Tasks

Stephen Pasteris*, Shiqiang Wang†, Christian Makaya†, Kevin Chan‡ and Mark Herbster*
*University College London (UK), †IBM Research (US), ‡Army Research Laboratory (US)

***Abstract*—We consider a distributed analytics system with interconnected machines. Analytics tasks run on the machines, where each task runs on a single machine but may require data from multiple other machines. Every task requires a given amount of data to run, and it needs to receive all its data within a specific deadline. The application scenario is that each machine has limited storage, thus we usually cannot place the entire amount of data for a specific task on a single machine that executes the task. We study how to distribute the data on machines in the system, without violating the bandwidth and storage constraints, while ensuring that the data transfer deadlines are met. We prove that a solution to this problem is equivalent to that of a max-flow problem on a specifically constructed graph. We present an algorithm for solving this problem via standard max-flow algorithms.**

I. Problem Formulation

We have a complete directed graph on a set of vertices V. The vertices of the graph represent machines that can store and process data. The machines are connected by communication pipelines (edges) that carry data. Each edge (i, j) has a weight $B(i, j)$ which represents the bandwidth of the communication pipeline from i to j, measured in bits per second. If there is no communication pipeline between i and j, then $B(i, j) = 0$. The bandwidth is measured in bits per second and is the maximum rate that information can be sent between two machines. We assume that every communication pipeline has zero latency. Each machine i has a weight S_i which represents the maximum amount of information (called the "storage capacity") that can be stored in machine i. If a machine i is a router without storage capability, then we have $S_i = 0$. We are given a set W of "tasks". Every task k has an associated machine $a(k)$ and a weight D_k. Each task k represents an application that is run on machine $a(k)$ and requires D_k bits of information to run. With every task k we are given a maximum time, T_k, in which we need to send all D_k bits of information to machine $a(k)$. The problem is to find a distribution of the data across all machines that satisfies this constraint, given that no two tasks are called at the same time.

A. Information Streams

A propagation of information around the network is formally defined as an "information stream" which is a function $f : V \times V \to \mathbb{R}^+$, where $\mathbb{R}^+$ denotes the set of non-negative real numbers, satisfying $f(i, j) \leq B(i, j)$ for all machines i and j. Intuitively, the value $f(i, j)$ is the rate of information being transmitted from machine i to machine j, and the constraints mean that the rate of information being transmitted from one machine to another is no more than the bandwidth of the communication pipeline between them.

Given an information stream f and a machine i, the "emission rate" $l_f(i)$ is defined as $l_f(i) := \sum_{j \neq i} f(i, j) - \sum_{j \neq i} f(j, i)$, which is the amount of information leaving the machine minus the amount of information entering the machine. Intuitively, the emission rate is the rate of information being sent from a machine (if the emission rate is negative then the machine is receiving information).

Given a machine i, an "i-targeted information stream" is an information stream f satisfying $l_f(i) \leq 0$ and $l_f(j) \geq 0$ for all $j \neq i$. Intuitively an i-targeted information stream is an information stream in which machine i receives information and all other machines send information. Let Ω_i be the set of all i-targeted information streams.

B. Data Distribution and Task Time

Let W be the set of tasks. Given a task k, let D_k be the quantity of data that it requires and let $a(k)$ be the machine that it is run on. Our algorithm will find a distribution of the data across the network, which is a function $d : W \times V \to \mathbb{R}^+$ where $d(k, i)$ is the amount of data for task k which is stored at machine i. This implies the following constraints: $\sum_i d(k, i) = D_k$, i.e., the total amount of data for task k stored across the network is equal to D_k; and $\sum_k d(k, i) \leq S_i$, which means the total amount of data stored at a machine is no more than the storage capacity at that machine.

Given a task k, and an $a(k)$-targeted information stream $f_k \in \Omega_{a(k)}$, the "task time" is equal to $\max_{i \in V \setminus \{a(k)\}} d(k, i)/l_{f_k}(i)$ which is the maximum of time for the required data to leave a given machine and is hence the time taken to propagate the data to machine $a(k)$ (as the communication pipelines have zero latency).

C. The Problem

With each task we are given the machine it runs on, $a(k)$, the quantity of data that it requires, D_k, as well as a maximum acceptable task time T_k. With each machine i we are given a storage capacity S_i. The problem is to find a distribution $d(\cdot, \cdot)$ (as defined in the above section), and $a(k)$-information streams such that the task time of every task is no greater than its maximum acceptable task time. Formally, this problem is as follows:

Find a function $d : W \times V \to \mathbb{R}^+$ and, for every $k \in W$, an $a(k)$-targeted information stream $f_k \in \Omega_{a(k)}$ such that the following constraints are met:

1) For all $k \in W$, $\sum_{i \in V} d(k, i) = D_k$
2) For all $i \in V$, $\sum_{k \in W} d(k, i) \leq S_i$
3) For all $k \in W$, $\max_{i \in V \setminus \{a(k)\}} d(k, j)/l_{f_k}(i) \leq T_k$

The full version of this paper has been published in [1].

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy-right notation hereon.

II. Algorithm

We now outline the algorithm for solving the above problem. The algorithm works by converting the problem into a max-flow problem [2] as follows:

1) Construct a directed graph G (with capacities on edges) as follows (note that we will refer to vertices of the original graph as "machines" and vertices and edges of G as "G-vertices" and "G-edges"):
    a) For every machine i and every task k create a G-vertex $v(i, k)$.
    b) For every machine i create a G-vertex $w(i)$.
    c) Create a source G-vertex s and a sink G-vertex t.
    d) For all machines i and j and tasks k, add a G-edge from $v(i, k)$ to $v(j, k)$ with capacity $T_k B(i, j)$.
    e) For all machines i add a G-edge from s to $w(i)$ with capacity S_i.
    f) For all machines i and tasks k add a G-edge from $w(i)$ to $v(i, k)$ with infinite capacity.
    g) For all tasks k add a G-edge from $v(a(k), k)$ to t with capacity D_k.
2) Find a maximum flow from s to t. Given G-vertices x and y, let F_{xy} be the flow from x to y.
3) If there exists a task k with $F_{v(a(k),k)t} < D_k$ then a required data distribution does not exist, i.e., the problem is not feasible. Else, the required data distribution is found by setting $d(k, i) := F_{w(i)v(i,k)}$ for all $i \in V$ and $k \in W$.

III. Proof of Correctness

A. Equivalent Flows

Given a task k and an $a(k)$-targeted information stream $f_k \in \Omega_{a(k)}$ with task time at most T_k, we define the function $\hat{f}_k : V \times V \to \mathbb{R}^+$ as follows: Given machines i and j, let $\hat{f}_k$ be the total amount of information relevant to the task (i.e., that stored in the machines) which passes from i to j when the information for task k is transferred to $a(k)$ via the information stream f_k. For all $i, j \in V$, we clearly have $\hat{f}_k(i, j) \leq T_k B(i, j)$ as this is the maximum amount of information that can be passed from i to j in time T_k. For all $i \neq a(k)$ we have that $\sum_j \hat{f}_k(i, j) - \sum_j \hat{f}_k(j, i)$ is the total amount of relevant information sent from machine i which is equal to $d(k, i)$. We also have that $\sum_j \hat{f}_k(j, a(k)) - \sum_j \hat{f}_k(a(k), j)$ is the total amount of relevant information entering machine $a(k)$ which is equal to $D_k - d(k, a(k))$. So, to summarise we have that $\hat{f}_k$ satisfies the following conditions:

1) $\hat{f}_k(i, j) \leq T_k B(i, j)$ for all $i, j \in V$
2) $\sum_j \hat{f}_k(i, j) - \sum_j \hat{f}_k(j, i) = d(k, i)$ for all $i \neq a(k)$
3) $\sum_j \hat{f}_k(j, a(k)) - \sum_j \hat{f}_k(a(k), j) = D_k - d(k, a(k))$

We now show the converse: that given an $\hat{f}_k$ satisfying the above conditions, there is an $a(k)$-information stream, f_k, with task time at most T_k. This is found by setting $f_k(i, j) := \hat{f}_k(i, j)/T_k$ for all $i, j \in V$ since, directly from the above respective conditions, we have:

1) $f_k(i, j) \leq B(i, j)$ for all $i, j \in V$
2) $l_{f_k}(i) = d(k, i)/T_k \geq 0$ for all $i \neq a(k)$
3) $l_{f_k}(a(k)) = -(D_k - d(k, a(k)))/T_k \leq 0$

so f_k is an $a(k)$-targeted information stream. Note that, by above, the task time is equal to $\max_{i \in V \setminus \{a(k)\}} d(k, i)/l_{f_k}(i) = T_k$. We have hence shown that an $a(k)$-targeted information stream with task time at most T_k exists if and only if an $\hat{f}_k$ exists that satisfies the above conditions.

B. Existence of Flow

We now turn to the weighted directed graph G that the algorithm constructs. We first show that, given a required distribution $d(\cdot, \cdot)$ exists, then any maximal flow F from s to t has $\sum_k F_{v(a(k),k)t} = \sum_k D_k$. First note that since the sum of the capacities of the G-edges entering t is equal to $\sum_k D_k$ the flow can never be greater than this value. Hence, all that is required is to construct a flow, F, with $\sum_k F_{v(a(k),k)t} = \sum_k D_k$. We now construct such a flow. First define $F_{w(i)v(i,k)} := d(k, i)$ for all $i \in V$ and $k \in W$. Define $F_{s,w(i)} := \sum_k d(k, i)$. Given a task k choose $\hat{f}_k$ as defined in the previous subsection. We then define $F_{v(i,k)v(j,k)} := \hat{f}_k(i, j)$ for all $i, j \in V$, and define $F_{v(a(k),k)t} := D_k$. We clearly have $\sum_k F_{v(a(k),k)t} = \sum_k D_k$. We now show that F satisfies the conditions of a flow:

1) For all $i \in V$: $F_{sw(i)} - \sum_k F_{w(i)v(i,k)} = 0$.
2) For all $k \in W$ and $i \in V \setminus \{a(k)\}$: $F_{w(i)v(i,k)} + \sum_{j \neq i} F_{v(j,k)v(i,k)} - \sum_{j \neq i} F_{v(i,k)v(j,k)} = 0$.
3) For all $k \in W$: $F_{w(a(k))v(a(k),k)} + \sum_{j \neq a(k)} F_{v(j,k)v(a(k),k)} - \sum_{j \neq a(k)} F_{v(a(k),k)v(j,k)} - F_{v(a(k),k)t} = 0$.
4) For all $i \in V$: $F_{sw(i)} = \sum_k d(k, i) \leq S_i$.
5) For all $i, j \in V$ and $k \in W$: $F_{v(i,k)v(j,k)} = \hat{f}_k(i, j) \leq T_k B(i, j)$.

which proves the existence of such a flow.

C. Sufficiency of Flow

We now show that given any flow F from s to t in G, with $\sum_k F_{v(a(k),k)t} = \sum_k D_k$, we have a sufficient distribution $d(\cdot, \cdot)$ defined as $d(k, i) := F_{w(i)v(i,k)}$. First note that since $\sum_k F_{v(a(k),k)t} = \sum_k D_k$ which is the maximum amount of flow that can enter t, we must have that every G-edge $(v(a(k), k), t)$ is at full capacity: i.e., $F_{v(a(k),k)t} = D_k, \forall k$.

Given $k \in W$, let G_k be the subgraph of G induced by the G-vertices $\{v(i, k) : i \in V\}$. Note that the only G-edges entering G_k are $\{(w(i), v(i, k)) : i \in V\}$ and the only G-edge exiting G_k is $(v(a(k), k), t)$. Since F is a flow, this implies that $\sum_i d(k, i) = \sum_i F_{w(i)v(i,k)} = F_{v(a(k),k)t} = D_k$ as required.

Note that for all $i \in V$ we have, since F is a flow, that $S_i \geq F_{sw(i)} = \sum_k F_{w(i)v(i,k)} = \sum_k d(k, i)$ as required.

For all $k \in W$ define $\hat{f}_k : V \times V \to \mathbb{R}^+$ as $\hat{f}_k(i, j) := F_{v(i,k)v(j,k)}$. We now show that the above conditions (in Section III-A) for $\hat{f}_k$ are satisfied:

1) For all $i, j \in V$ we have $F_{v(i,k)v(j,k)} \leq T_k B(i, j)$.
2) For all $i \neq a(k)$ we have $\sum_{j \neq i} \hat{f}_k(i, j) - \sum_{j \neq i} \hat{f}_k(j, i) = d(k, i)$.
3) We have $\sum_{j \neq a(k)} \hat{f}_k(j, a(k)) - \sum_{j \neq a(k)} \hat{f}_k(a(k), j) = D_k - d(k, a(k))$.

which, as shown above, implies the existence of an $a(k)$-targeted information stream with task time T_k.

For full details, please refer to the full paper at [1].

References

[1] S. Pasteris, S. Wang, C. Makaya, K. Chan, and M. Herbster, "Data distribution and scheduling for distributed analytics tasks," in *DAIS Workshop*, 2017. [Online]. Available: https://dais-ita.org/node/854

[2] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, no. 2, pp. 248–264, Apr. 1972.