The Resource Allocation Problem for CERN Dataset Transfer

Qiao Xiang
qiao.xiang@cs.yale.edu

1 The Offline Model for Resource Allocation Problem

1.1 Flow prioritized by bandwidth constraint

We use a directed graph $G = (V, E)$ to denote the network. And we consider the system operating in a discrete-time, where time is divided into time slots with equal length and indexed as $t = 1, 2, \ldots, T$. Given a time slot t, each link (i, j) have a link capacity denoted as $u_{ij}(t)$. At the beginning of time slot 1, we are given a set of dataset transfer requests, denoted as $\mathbb{M}$. Each request $m \in \mathbb{M}$ is associate with a priority level p_m. This priority p_m indicates that the transmission rate per time slot of m must be between a predefined interval $[L(p_m), U(p_m)]$ before m is accomplished, with the only exception in the final transmission slot where $L(p_m)$ may be violated. With this setting, a dataset transfer request m can be represented as a 5-tuple $\{s_m, d_m, v_m, L(p_m), U(p_m)\}$. In this tuple, s_m denotes the source location of dataset, d_m denotes where the dataset should be sent to, and v_m denotes the volume of dataset.

Decision Variables: For each request m, we have three sets of decision variables in the model:

- T_m, the number of time slots needed to finish the transfer of m;
- $r_m(t)$, the data volume of data transfer request m that is transferred in time slot t;
- $f_{ij}^m(t)$, the data volume of data transfer m along link $(i, j) \in E$ in time slot t.

And we formulate the resource allocation problem (**RAP**) as the following model.

$$\textbf{RAP} \quad \text{minimize} \quad T \tag{1}$$

subject to

$$\sum_m f_{ij}^m(t) \le u_{ij}(t) \qquad \forall (i,j) \in E \text{ and } t, \tag{2a}$$

$$\sum_{j \in V} f_{ij}^m(t) - \sum_{j \in V} f_{ji}^m(t) = 0 \qquad \forall m, t \text{ and } i \in V/\{s_m, d_m\}, \tag{2b}$$

$$\sum_{j \in V} f_{ij}^m(t) - \sum_{j \in V} f_{ji}^m(t) = r_m(t) \qquad \forall m, t \text{ and } i = s_m, \tag{2c}$$

$$\sum_{j \in V} f_{ij}^m(t) - \sum_{j \in V} f_{ji}^m(t) = -r_m(t) \qquad \forall m, t \text{ and } i = d_m, \tag{2d}$$

$$\sum_{t=1}^{T_m} r_m(t) = v_m \qquad \forall m \tag{2e}$$

$$r_m(t) \ge L(p_m) \qquad \forall m \text{ and } t < T_m, \tag{2f}$$

$$r_m(t) = 0 \qquad \forall m \text{ and } t > T_m, \tag{2g}$$

$$T_m \le T \qquad \forall m, \tag{2h}$$

$$r_m(t) \le U(p_m), \qquad \forall m \text{ and } t, \tag{2i}$$

$$f_{ij}^m(t) \ge 0 \qquad \forall (i,j) \in E, m, \tag{2j}$$

RAP aims to minimize the maximum transfer time slot T of all data transfer requests. In this model, Constraint (2a) ensures the total assigned data rate on each link does not exceed the link capacity. Con-

straints (2b)-(2d) represent the flow conservation. Constraints (2e)-(2i) ensure that during the data transfer, the transmission rate of each request m satisfies its predefined rate upper bound and lower bound. We see that even with equal-priority case, finding such T is very challenging. To this end, we next develop a max-min fair resource allocation algorithm for this problem.

1.2 A Max-Min Fair Resource Allocation Algorithm

To present the max-min resource allocation (MFRA) algorithm, we first define the Max-Min Fair $\mathbf{MMF}$ problem for each time slot t.

$$\mathbf{MMF}(t) \quad \text{maximize} \quad z(t) \tag{3}$$

subject to

$$\sum_{m} f_{ij}^{m}(t) \leq u_{ij}(t) \qquad \forall (i,j) \in E \tag{4a}$$

$$r_m(t) \geq z(t) \cdot v_m(t) \qquad \forall m \in M_{unsat}(t) \tag{4b}$$

$$r_m(t) = z_{sat}^{m}(t) \cdot v_m(t) \qquad \forall m \in M_{sat}(t) \tag{4c}$$

$$\sum_{j \in V} f_{ij}^{m}(t) - \sum_{j \in V} f_{ji}^{m}(t) \begin{cases} = r_m(t) & \text{if } i = s_m \\ = 0 & \text{if } i \in V/\{s_m, d_m\} \\ = -r_m(t) & \text{if } i = d_m \end{cases} \qquad \forall m \in M(t), \tag{4d}$$

$$r_m(t) \geq min(L(p_m), v_m - \sum_{\tau=1}^{t-1} r_m(\tau)), \qquad \forall m \in M(t), \tag{4e}$$

$$r_m(t) \leq U(p_m), \qquad \forall m \in M(t), \tag{4f}$$

$$z(t) \geq 0 \tag{4g}$$

Then we present the MFRA algorithm as shown in Algorithm 1. The basic idea of MFRA is to iteratively maximize the minimum data transfer satisfaction rate until all the requests are saturated, i.e., achieving maximum $z(t)$. In each iteration, newly saturated requests are removed from the subsequent process by fixing their corresponding rate value. In particular, it first solves the **MMF** problem (Line 6). Then each unsaturated transfer request m goes through the saturation test (Line 8-15). In this test, MFRA first checks for residual paths in the network for request m (Line 8). If there is no such path, a further test is performed by solving **MMF** with a new setting, in which $M_{unsat}(t)$ only contains request m, and all other requests which were in $M_{unsat}(t)$ are moved to $M_{sat}(t)$ with a satisfaction rate $z(t)$ (Line 9). If the new $z^{temp}(t)$ is the same as $z(t)$, it means that the request m is indeed saturated. We then update both $M_{sat}(t)$ and $M_{unsat}(t)$ (Line 10-14). After all data transfers are saturated in time slot t, we can derive a feasible set of $f_{ij}^{m}(t)$ for each transfer request m, and the corresponding $r_m(t)$ (Line 22-23). If the total saturation rate of a transfer request m is 1, it means that this request has been fulfilled, and we can remove it from the unfinished set of requests (Line 23-25), which will be pushed for next time slots. In this way, MFRA achieves a balance between fairness of data transfers and network utilization. Furthermore, it has a low computational complexity.

Note: Here we allocate not only bandwidth, i.e., scheduling, but also routes to different transfer requests. However, MFRA can certainly handle the case where particular routing constraints are specified, e.g., all routes are fixed ahead or each transfer request only uses one single path in each time slot, by adding an additional set of linear constraints in the formulation of **MMF** problem.

Algorithm 1 MFRA: a Max-Min Fair Resource Allocation Algorithm for the **RAP** problem

$t \leftarrow 0$
while $\mathbb{M} \neq \emptyset$ **do**
  $t \leftarrow t + 1$
  $M_{sat}(t) \leftarrow \emptyset, M_{unsat}(t) \leftarrow \mathbb{M}_p$
  while $M_{unsat}(t) \neq \emptyset$ **do**
    solve $\mathbf{MMF}(t)$ and get the optimal value $z(t)$
    for each $m \in M_{unsat}(t)$ **do**
      if there is no residual path for transfer request m to accommodate more flows **then**
        solve $\mathbf{MMF}(t)$ with revised inputs $M_{unsat}^{temp}(t) \leftarrow m$, $M_{sat}^{temp}(t) \leftarrow M_{sat}(t) \cup M_{unsat}(t)/\{m\}$ and $z_{sat}^{m}(t) \leftarrow z(t)$ for any $m \in M_{unsat}(t)/\{m\}$, and get the optimal value $z^{temp}(t)$
        if $z^{temp}(t) == z(t)$ **then**
          $M_{sat}(t) \leftarrow M_{sat}(t) \cup \{m\}$
          $M_{unsat}(t) \leftarrow M_{unsat}(t)/\{m\}$
          $z_{sat}^{m}(t) \leftarrow z(t)$
        end if
      end if
    end for
  end while
  for each $m \in M_{sat}(t)$ **do**
    if $\sum_{t_0=1}^{t} z_{sat}^{m}(t_0) \geq 1$ **then**
      $z_{sat}^{m}(t) \leftarrow 1 - \sum_{t0=1}^{t-1} z_{sat}^{m}(t0)$
    end if
    Derive a feasible set of $f_{ij}^{m}(t)$ and the corresponding $r_m(t)$ for each data transfer request $m \in \mathbb{M}(t)$ based on constraints (4a)-(4g)
    if $\sum_{t_0=1}^{t} z_{sat}^{m}(t_0) == 1$ **then**
      $\mathbb{M} \leftarrow \mathbb{M}/\{m\}$
    end if
  end for
end while
$T \leftarrow t$
return T