ProjecToR: Agile Reconfigurable Data Center Interconnect

Abstract

This supplementary document describes the scheduling algorithm used in the ProjecToR data-center interconnect system. The algorithm assigns bundles to laser-photodetector edges as they arrive *online* with the goal of minimizing the bundle/flow completion times over all bundles in the system. It does this while taking advantage of the reconfigurability of the network and establishing edges between laser and photodetector pairs as necessary.

We compare the performance of the proposed stable-matching algorithm to the optimal performance of a more powerful hindsight optimal algorithm that is aware of all the bundles arriving in the system and can operate offline. We can show that for any ϵ, if the proposed stable-matching algorithm is allowed to run $(2+\epsilon)$ faster than the hindsight optimal algorithm, then it can achieve a total weighted bundle latency that is $\left(\frac{2}{\epsilon}+1\right)$-competitive with the hindsight optimal algorithm. The proof uses a dual-fitting technique, where the objective function is bounded by finding an appropriate feasible solution for the dual problem, to give the bound.

We also give an integer linear programming formulation for the problem of maximizing instantaneous throughput.

1 Problem formulation and main result

Problem definition: We are given a graph G with 4 sets of vertices: sources S, destinations D, lasers (transmitters) T, and photodetectors (receivers) R. Each laser in T is attached to a particular source in S, and each photodetector in R is attached to a particular destination in D. The edges in the graph $E(G)$ are all directed and go from S to T to R to D.

Bundles j from set J arrive *online*. Say each bundle j arrives at time r_j, and is specified by a source destination pair (s_j, d_j). We say that a bundle j can be transmitted on edge $e := (u, v)$ with $u \in T, v \in R,$, if there are edges (s_j, u) and (v, d_j) in $E(G)$. We denote this by $j \sim e$. A bundle also has a weight w_j. This weight signifies the relative importance of the bundles. Finally, assume the length of each bundle is given by p_j (an integer).

Time is discrete, and is indexed by $t = 1, 2, \ldots$. We assume a synchronous model where clocks on all the vertices tick at the same time. At any time t, each *active* edge e can be used to transmit one unit of bundle $j \sim e$ released earlier (i.e., $r_j \leq t$). As a physical constraint, each transmitter or receiver can only be adjacent to only *one* active edge at a time. A *scheduling algorithm*, for each time t, picks the set of active edges and the bundles that are transmitted on each active edge.

The completion time, c_j, of a bundle is the time at which all p_j units of the bundle j have been transmitted. The latency (or flow time) of a bundle is the time it stays in the system, i.e., time $c_j - r_j$. The objective of the scheduling algorithm is to minimize the weighted sum of latencies over all bundles, i.e. to minimize

$$\sum_j w_j(c_j - r_j).$$

Another objective is to minimize the ℓ_2 objective:

$$\sum_j w_j(c_j - r_j)^2.$$

Competitive analysis compares the cost of an algorithm to that of a hindsight optimal solution. The hindsight optimal solution is aware of all future bundle arrivals and schedules them optimally. An algorithm is *α-competitive* if its cost is at most an α multiplicative factor away from the hindsight optimal cost.

For some problems such as the one we consider, we also need *speed augmentation*: algorithm runs at a speed that is a constant β multiplicative factor faster than the hindsight optimal solution.

We assume, for simplicity, that each bundle j is of unit length and has weight w_j. A bundle of length p_j and weight w_j can be treated as p_j unit length bundles each with weight $\frac{w_j}{p_j}$, but this changes the objective and gives what is known as the fractional flow time. Guarantees about fractional flow time can be converted to guarantees about flow time using standard techniques.

Stable allocation: An *allocation* is a map $A : J \to E(G) \cup \{\perp\}$ such that $\forall j \in J, A(j) \neq \perp \Rightarrow j \sim A(j)$, and $Im(A)$ forms a matching; it specifies a feasible allocation of bundles to active edges. Bundles with image $\perp$ are not allocated to any edge.

Given priorities, $\pi_j\ \forall j \in J$, an allocation A is a *stable allocation* iff

$$\forall j : A(j) = \perp, \forall e \sim j, \exists j' : A(j') \text{ and } e \text{ share a common vertex, and } \pi_{j'} \geq \pi_j.$$

A stable allocation is one where a bundle is not blocked by lower priority bundles. For every feasible edge of an unallocated bundle, either of the end points of the edge must be transmitting a higher priority bundle. Stable allocations always exist, but need not be unique.

The stable allocation scheduling algorithm is as follows: at any given time, among all bundles that have arrived but not completely transmitted, transmit a set of bundles corresponding to any stable allocation.

Theorem 1. *For any $\epsilon \in (0, 1)$, for instances with with unit length bundles, the stable allocation scheduling algorithm is $(2 + \epsilon)$-speed $\frac{2}{\epsilon} + 1$-competitive for the objective of minimizing*

- *total weighted latency of the bundles, by using the priority $\pi_j = w_j$.*
- *the ℓ_2 objective, by using the priority $\pi_j = w_j(t - r_j)$.*

The result also extends to the average latency of bundles for any finite run of the algorithm.

2 Analysis

We prove the theorem for the weighted sum objective; the proof for the ℓ_2 objective is similar. We use the dual-fitting technique, a standard method used in approximation and online algorithms literature, to analyze our algorithm. The dual fitting analysis technique consists of two main steps:

1. Formulate the underlying optimization problem as an integer linear program, and then relax the constraints to obtain a linear program. This is called the LP relaxation of the problem. An optimal solution to the linear program gives a lower bound on the cost of optimal solution of the original optimization problem.

2. Take the dual of the linear program and appropriately set the dual variables to bound the cost of the algorithm.

We start by writing the LP relaxation. The variables x_{jet} in (1) denote the fraction of bundle j that is scheduled for transmission on edge e at time t. The first set of constraints enforce that every bundle is scheduled at some time instant t. The second and third set of constraints ensure that at each time instant t, no transmitter or receiver has more than one outgoing edge.

We incorporate speed augmentation by restricting the hindsight optimal algorithm: it can only transmit $\frac{1}{2+\epsilon}$ fraction of a bundle in one time step (for some $\epsilon > 0$). This is equivalent to providing our algorithm with a speed up of $2+\epsilon$. The objective function is a valid lower bound on the optimal value, since a bundle j that is scheduled at time t has a latency precisely $t - r_j + 1$. Our relaxation is based on similar LP relaxations used in [1].

$$\text{Minimize} \sum_j \sum_{e \in G} w_j \cdot x_{jet} \cdot (t - r_j + 1) \tag{1}$$

$$\begin{aligned}
s.t. \quad \sum_{t \geq r_j} x_{jet} &\geq 1 && \forall j \\
\sum_{j:e->v} x_{jet} &\leq \frac{1}{2+\epsilon} && \forall v \in T, \forall t \\
\sum_{j:e->u} x_{jet} &\leq \frac{1}{2+\epsilon} && \forall v \in R, \forall t \\
x_{jet} &\geq 0 && \forall e, j, t
\end{aligned}$$

Next, we write the dual program. The dual program has variables α_j corresponding to the first set of primal constraints. Variables β_{ut}, β_{vt} correspond to the second and third set of constraints.

$$\text{Maximize} \sum_j \alpha_j - \frac{1}{2+\epsilon} \cdot \left(\sum_{u \in T} \beta_{ut} + \sum_{v \in R} \beta_{vt} \right) \tag{2}$$

$$\begin{aligned}
s.t \quad \alpha_j - \sum_{u:e->u} \beta_{ut} - \sum_{v:e->v} \beta_{vt} &\leq w_j \cdot (t - r_j + 1) && \forall e, t, j-> e && (3) \\
\alpha_j &\geq 0 && \forall j \\
\beta_{ut}, \beta_{vt} &\geq 0 && \forall u \in T, \forall v \in R
\end{aligned}$$

The second step of the proof is to set the dual variables that can give a bound on the competitive ratio of our algorithm. Consider the schedule produced by our algorithm. Let $J(t)$ denote the set of bundles that are unfinished at time t. For a bundle j, define $J_{<j}(e, r_j)$ as the set of bundles that arrive earlier than j and are scheduled to be transmitted after j using either the transmitter or the receiver belonging to edge e. Similarly, define $J_{>j}(e, r_j)$ as the set of bundles that are released earlier than bundle j and are scheduled before j using either the transmitter or the receiver belonging to edge e.

Let t_j denote the time at which bundle j is transmitted and let $e(j)$ denote the edge on which it is transmitted.

Setting of Dual Variables: We set the dual variables α_j, β_e as follows.

- We set $\alpha_j = w_j \cdot (|J_{>j}(e(j), r_j)| + 1) + \sum_{j' \in J_{<j}(e(j),r_j)} w_{j'}$. Note that bundles in the set $J_{>j}(e(j), r_j)$ delay bundle j from the definition of our algorithm. (A bundle j can also get delayed by another bundle j' that arrives after it but we charge that increase to bundle j'). The second term, $\sum_{j' \in J_{<j}(e(j),r_j)} w_{j'}$ accounts for increase in the latency that bundle j causes for the other bundles. In particular, we only account for the increase to bundles that are released earlier than bundle j and use the same edge $e(j)$. From the definition, it is clear that α_j is does not change over time, and depends only the set of bundles that arrived before bundle j. This will be important to verify constraints.

- We set $\beta_{ut} = \sum_{j:j->u, j \in J(t)} w_j$ and $\beta_{vt} = \sum_{j:j->v, j \in J(t)} w_j$. In words, we set β_{ut}, β_{vt} to be the total weight of bundles that are unfinished at time t that are scheduled by our algorithm to use vertex u or vertex v.

Bounding the Dual objective: We show that our setting of dual variables capture the cost incurred by our algorithm.

Lemma 2. *$\sum_j \alpha_j = CostAlgo$.*

Proof. Define $J_{>j}(e, > r_j)$ as the set of bundles that arrive after j and are scheduled ahead of j using either the transmitter or the receiver belonging to edge e. From the definition of α_j variables and rearranging terms we get,

$$\begin{aligned}
\sum_j \alpha_j &= \sum_j (w_j \cdot (|J_{>j}(e(j), r_j)| + 1) + \sum_{j' \in J_{<j}(e(j),r_j)} w_{j'}) \\
&= \sum_j w_j \cdot (|J_{>j}(e(j), r_j)| + 1) + \sum_j \sum_{j' \in J_{<j}(e(j),r_j)} w_{j'} \\
&= \sum_j w_j \cdot (|J_{>j}(e(j), r_j)| + 1 + |J_{>j}(e, > r_j|)
\end{aligned}$$

Now, observe that $(|J_{>j}(e(j), r_j)| + 1 + |J_{>j}(e, > r_j)|)$ is the amount of time bundle j spends before being transmitted. Therefore, $\sum_j \alpha_j$ is precisely the total latency of bundles in our algorithm. □

Next, we prove that β variables also account for the cost algorithm.

Lemma 3. *$\sum_{u,t} \beta_{ut} = \sum_{v,t} \beta_{vt} = CostAlgo$.*

Proof. Fix a bundle j and consider the interval $[r_j, t_j]$; recall that r_j and t_j denote the arrival time and transmission time of bundle j. From the definition of β_{ut} and β_{vt} variables, j contributes w_j to each time instant $t \in [r_j, t_j]$. Therefore, the lemma follows. □

The next theorem follows immediately from the two lemmas above.

Theorem 4. *The cost of dual (2) is at least $\frac{\epsilon}{2+\epsilon} CostAlgo$.*

Proof. Consider the objective function of dual program (2). From Lemmas 2, 3 we get,

$$\sum_j \alpha_j - \frac{1}{2+\epsilon} \cdot \left(\sum_{u \in T} \beta_{ut} + \sum_{v \in R} \beta_{vt} \right) = \text{CostAlgo} - \frac{2 \text{ CostAlgo}}{2+\epsilon} = \frac{\epsilon}{2+\epsilon} \text{CostAlgo}.$$

□

Verifying Constraints: To complete the proof of Theorem 1, it remains to show that our choice of dual variables satisfies the dual constraints. Fix a bundle j between (s, d). There is a dual constraint 3 for every edge e on which bundle j can be transmitted and for each time instant t. Fix an edge e and time instant t'. Rearranging 3, we need to show that

$$\alpha_j \leq \sum_{u:e->u} \beta_{ut} + \sum_{v:e->v} \beta_{vt} + w_j \cdot (t' - r_j + 1).$$

Since our setting of α_j depends only on the set of bundles that arrived earlier than j, we assume that no more bundles enter the system. This is without loss generality since the arrival of more bundles can only increase the value of the β variables compared to the case when no more bundles arrive. Consider,

$$\begin{aligned} \alpha_j &= w_j \cdot (|J_{>j}(e(j), r_j)| + 1) + \sum_{j' \in J_{<j}(e(j), r_j)} w_{j'} \\ &\leq \sum_{j'' \in J_{>j}(e(j), r_j} w_{j''} + \sum_{j' \in J_{<j}(e(j), r_j)} w_{j'} + w_j \cdot (t' - r_j) \end{aligned} \tag{4}$$

The last inequality follows from two facts:

1. Every bundle $j'' \in J_{>j}(e(j), r_j)$ has weight higher than weight of bundle j;
2. At each time instant $t \in [r_j, t']$ the bundle j is blocked at the transmitters or at the receivers by a bundle of higher weight.

Observe, however, that all the bundles in the sets $J_{>j}(e(j), r_j)$ and $J_{<j}(e(j), r_j)$ are alive at time t' and contribute their weight β variables. Therefore,

$$\sum_{j'' \in J_{>j}(e(j), r_j} w_{j''} + \sum_{j' \in J_{<j}(e(j), r_j)} w_{j'} + w_j \cdot (t' - r_j) \leq \sum_{u:e->u} \beta_{ut} + \sum_{v:e->v} \beta_{vt} + w_j \cdot (t' - r_j) \tag{5}$$

From equations (4,5) we conclude that all the dual constraints are satisfied. This completes the proof.

3 Maximizing Instantaneous Throughput

In this section, we give an integer linear program for the problem of maximizing instantaneous throughput. The problem is an instantaneous version of the one defined in Section 1: we are given the two-tier network G and a set of bundles J, and we need to find an allocation A of jobs to a set of active edges to maximize the weighted throughput, which is the sum of the weights of the jobs that are allocated:

$$w(A) := \sum_{j:A(j)\neq\bot} w_j.$$

The integer linear program (ILP) has a variable x_{je} for each bundle j and each edge $e \sim j$. Setting $x_{je} = 1$ indicates that e is active and $A(j) = e$; conversely, x_{je} is set to 0 otherwise.

$$\text{Maximize} \sum_{j,e} w_j \cdot x_{je} \tag{6}$$

$$\begin{aligned}
\sum_{e:j\sim e} x_{je} &\leq p_j && \forall j \\
\sum_{j,e:j\sim e=(u,v)\in E} x_{je} &\leq 1 && \forall u \in T \\
\sum_{j,e:j\sim e=(u,v)\in E} x_{je} &\leq 1 && \forall v \in R \\
x_{je} &\geq 0 && \forall e, j.
\end{aligned}$$

We give a brief explanation of the ILP. Objective function is straight word and maximizes the total weight of bundles allocated. The first set of constraints make sure for every j, the total flow sent over all edges is at most p_j. The second and third constraints enforce the matching constraints on transmitters and receivers. Without the first set of constraints, the ILP is same as max-weight matching ILP on bipartite graphs, which is known to be integral, and the problem polynomial time solvable. The question of whether this problem is in polynomial time (P) or NP-Hard remains open.

References

[1] S Anand, Naveen Garg, and Amit Kumar. Resource augmentation for weighted flow-time explained by dual fitting. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1228–1241. SIAM, 2012. 2