

# Real-time GDA

## ABSTRACT

### ACM Reference format:

. 2016. Real-time GDA. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17)*, 2 pages.  
DOI: 10.475/123\_4

## 1 SINGLE DAG QUERY

### 1.1 Model

**1.1.1 Wide Area Network.** Let  $S$  be the set of sites that hold data and run tasks, and  $\mathcal{L}$  be the set of directed edges that represent inter-site links. For each inter-site WAN link  $(i, j) \in \mathcal{L}$ , let  $B_{ij}$  be the bandwidth and  $C_{ij}$  be the cost to transfer one unit of data from site  $i$  to site  $j$ .

**ASSUMPTION 1.** *The bandwidths are stable within the time frame of real-time data analytics.*

For the analysis we make the following assumption:

**ASSUMPTION 2.** *Data transfers on a particular link are non-overlapping. (To do: Show that this assumption does not add suboptimality to our objective, i.e., any overlapping schedule can be transformed into a non-overlapping schedule without any loss to the optimal value.)*

**1.1.2 DAG of tasks.** We define a stage in the DAG as a group of tasks that have the same input data dependencies. Let  $\mathcal{T}$  be the set of stages and  $\mathcal{D}$  be the directed set of edges that represent stage dependencies for the DAG, respectively. Some of the stages do not have any incoming edges and so represent the locations of the raw input data. Let  $\mathcal{R} \subset \mathcal{T}$  be the set of stages for the raw input data. The final stage in the DAG does not have any outgoing edges and so represent the final destination of the query's response denoted by  $F \in \mathcal{T}$ . Each stage dependency  $(k, l) \in \mathcal{D}$  also has a corresponding amount of data  $D^{kl}$  that must be transferred from stage  $k$  to stage  $l$ .

**ASSUMPTION 3.** *A stage must have completely received all of its input data before it can process and start transferring data to another stage.*

The DAG also has a start time  $T_0$  and a finish time  $T_f$  for which the schedule of data transfers must respect.

**1.1.3 Stage assignment decisions.** We model each stage as a group of tasks that must be scheduled to run together at the same site. The decision of stage  $k$  to be placed at site  $i$  is represented by the binary decision variable  $x_i^k \in \{0, 1\}$ . This means that for any directed edge  $(k, l) \in \mathcal{D}$ , then  $D^{kl}x_i^kx_j^l$  of data will be transferred across link  $(i, j) \in \mathcal{L}$ . Note that the set of stages  $\mathcal{R}$  and  $F$  which respectively represent the raw data and the DAG's final stage have decision variables which are preset according to its site-wise distribution and final stage location.

## 1.2 Stage Assignment Problem

**1.2.1 Problem Statement.** Given the available links for each dependency we have the following problem to minimize the cost of the WAN by deciding where to place each stage:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{(k,l) \in \mathcal{D}} \sum_{(i,j) \in \mathcal{L}} C_{ij} D^{kl} x_i^k x_j^l \\ \text{s.t.} \quad & \sum_{i \in S} x_i^k = 1 \quad \forall k \in \mathcal{T} \end{aligned} \quad (1a)$$

$$x_i^k \in \{0, 1\} \quad \forall i \in S, k \in \mathcal{T} \quad (1b)$$

Note that for any stage  $k \in \mathcal{R} \cup F$ , the placement decision is preset.

**1.2.2 Problem Reformulation.** Since  $x_i^k \in \{0, 1\}$  and  $x_j^l \in \{0, 1\}$ , then  $x_i^k x_j^l = 1$  iff  $\max\{0, x_i^k + x_j^l - 1\} = 1$ , and  $x_i^k x_j^l = 0$  iff  $\max\{0, x_i^k + x_j^l - 1\} = 0$ . This means that  $x_i^k x_j^l$  and  $\max\{0, x_i^k + x_j^l - 1\}$  are equivalent for the binary decisions and the later is a convex function where the former is not. Therefore, (1) can be reformulated with a convex objective function:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{(k,l) \in \mathcal{D}} \sum_{(i,j) \in \mathcal{L}} C_{ij} D^{kl} \max\{0, x_i^k + x_j^l - 1\} \\ \text{s.t.} \quad & \sum_{i \in S} x_i^k = 1 \quad \forall k \in \mathcal{T} \end{aligned} \quad (2a)$$

$$x_i^k \in \{0, 1\} \quad \forall i \in S, k \in \mathcal{T} \quad (2b)$$

The max function can be further simplified with the use of an auxiliary variable  $z_{ij}^{kl}$  to reformulate (2):

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{(k,l) \in \mathcal{D}} \sum_{(i,j) \in \mathcal{L}} C_{ij} D^{kl} z_{ij}^{kl} \\ \text{s.t.} \quad & \sum_{i \in S} x_i^k = 1 \quad \forall k \in \mathcal{T} \end{aligned} \quad (3a)$$

$$x_i^k \in \{0, 1\} \quad \forall i \in S, k \in \mathcal{T} \quad (3b)$$

$$z_{ij}^{kl} \geq x_i^k + x_j^l - 1 \quad \forall (i, j) \in \mathcal{L}, \forall (k, l) \in \mathcal{D} \quad (3c)$$

$$z_{ij}^{kl} \geq 0 \quad \forall (i, j) \in \mathcal{L}, \forall (k, l) \in \mathcal{D} \quad (3d)$$

If  $x_i^k$  can be relaxed from a binary variable to a nonnegative variable and still guarantee a binary optimal solution then (3) could be simply reformulated as a linear program.

**CONJECTURE 1.1.** *We can guarantee that relaxing the binary constraint (3b) to a nonnegativity constraint in (3) always has an optimal solution which is binary. We will prove this by proving that the constraints (3a) and (3c) form a totally unimodular matrix.*

## 1.3 Data Transfer Scheduling Problem

The goal is to find a schedule  $\pi$  given stage assignment  $\mathbf{x}$  that satisfies the stage dependencies  $\mathcal{D}$  from the DAG, the query's start time  $T_0$  and deadline  $T_f$ . Let  $t_0^{kl}$  and  $t_f^{kl}$  be the start and finish times to transfer data from stage  $k$  to stage  $l$ . Therefore, the schedule must

satisfy:

$$\max_{k:(k,l) \in \mathcal{D}} t_f^{kl}(\pi(\mathbf{x})) \leq \min_{m:(l,m) \in \mathcal{D}} t_0^{lm}(\pi(\mathbf{x})) \quad \forall l \in \mathcal{T} \quad (4a)$$

$$\min_{(k,l) \in \mathcal{D}} t_0^{kl}(\pi(\mathbf{x})) \geq T_0 \quad (4b)$$

$$\max_{(k,l) \in \mathcal{D}} t_f^{kl}(\pi(\mathbf{x})) \leq T_f \quad (4c)$$

The bandwidth constraints on each link must also be satisfied. Let  $d_{ij}^{kl}$  be the total amount of time (not necessarily contiguous) that link  $(i, j)$  is devoted to transfer data from stage  $k$  to stage  $l$ . Therefore, the schedule must satisfy:

$$d_{ij}^{kl} \geq \frac{D^{kl}}{B_{ij}} \quad \forall (i, j, k, l) : (i, j) \in \mathcal{L}, (k, l) \in \mathcal{D}, x_i^k = 1, x_j^l = 1 \quad (5)$$

CONJECTURE 1.2. [3] gives an optimal scheduling algorithm for a DAG on a single machine. With some effort, this can be proved for our case when the stage assignments have already been decided (multiple single machines with a DAG relating them all).

## 1.4 Joint optimization

The Stage Assignment Problem chooses the stage assignment to minimize WAN usage cost while the Data Transfer Scheduling Problem decides whether a stage assignment can be feasibly scheduled. Therefore, the joint optimization problem becomes finding a feasible stage assignment that minimizes the WAN usage cost.

CONJECTURE 1.3. *The Data Transfer Scheduling Problem can decide whether a given stage assignment has a feasible schedule or not. If infeasible, it also finds the sequence of bottlenecking data transfers that make it not satisfy the query deadline constraint. This infeasibility can be relayed back the Stage Assignment Problem's objective function with a carefully chosen dual cost.*

## REFERENCES

- [1] Dimitri P Bertsekas. 1999. *Nonlinear programming*. Athena scientific Belmont.
- [2] CE Gounaris and CA Floudas. 2008. Convexity of products of univariate functions and convexification transformations for geometric programming. *Journal of Optimization Theory and Applications* 138, 3 (2008), 407–427.
- [3] Eugene L Lawler. 1973. Optimal sequencing of a single machine subject to precedence constraints. *Management science* 19, 5 (1973), 544–546.
- [4] Ray PöRn, Kaj-Mikael Björk, and Tapio Westerlund. 2008. Global solution of optimization problems with signomial parts. *Discrete optimization* 5, 1 (2008), 108–120.

## APPENDIX