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 MODEL

Let $\mathcal{L}$ be the set of sites that holds data and runs tasks. For each inter-site WAN link, let B_i^j be the bandwidth from site $i \in \mathcal{L}$ to site $j \in \mathcal{L}$. We assume that the bandwidths are stable within the time frame of doing real-time data analytics.

1.1 Single MapReduce Query

We perform the map tasks at the sites that contain the associated data and denote D_i as the output data from all of the map tasks at site i. The fraction of reduce tasks assigned to site j is denoted as r_j which is also the fraction of all other sites' data that must be transfered through the WAN to j. This means that the total WAN usage is for a given task distribution r is:

$$\sum_i \sum_{j \neq i} D_i r_j \tag{1}$$

If we are given a start time s after the map steps are all completed and a data shuffle deadline t for the reduce tasks, then the completion time is bounded as such:

$$s + \max_{(i,j):j\neq i} \left\{ \frac{D_i r_j}{B_i^j} \right\} \leq t \tag{2}$$

which is caused by the heterogeneous WAN bandwidth and has a bottlenecking link(s). We assume that $t > s$.

2 PROBLEM FORMULATIONS

2.1 Single MapReduce Query

Minimize WAN usage. When minimizing WAN usage for a MapReduce data shuffle we have the following optimization problem:

$$\min_r \quad \sum_i \sum_{j \neq i} D_i r_j \tag{3a}$$

$$\text{s.t.} \quad \sum_j r_j = 1 \tag{3b}$$

$$r_j \geq 0 \quad \forall j \tag{3c}$$

Feasibility to meet deadline. We want to find a feasible r so that the data shuffle finishes at or before t:

$$\text{find } r \text{ s.t.} \quad \frac{D_i r_j}{B_i^j} \leq t - s \quad \forall (i,j) : j \neq i \tag{4a}$$

$$\sum_j r_j = 1 \tag{4b}$$

$$r_j \geq 0 \quad \forall j \tag{4c}$$

Minimize WAN usage given a shuffle deadline. When minimizing WAN usage for a MapReduce data shuffle for a given deadline t we have the following optimization problem:

$$\min_r \quad \sum_i \sum_{j \neq i} D_i r_j \tag{5a}$$

$$\text{s.t.} \quad \frac{D_i r_j}{B_i^j} \leq t - s \quad \forall (i,j) : i \neq j \tag{5b}$$

$$\sum_j r_j = 1 \tag{5c}$$

$$r_j \geq 0 \quad \forall j \tag{5d}$$

The objective function can be reformulated to:

$$\sum_i \sum_{j \neq i} D_i r_j = \sum_i D_i \sum_j r_j - \sum_j D_j r_j = \sum_i D_i - \sum_j D_j r_j \tag{6}$$

3 OPTIMA

3.1 Single MapReduce Query with a deadline

Since Problem (5) is a convex optimization problem (linear program), we look at the Karush-Kuhn-Tucker (KKT) conditions for optimality using the following dual variables $(\mu_{ij}, \theta, \lambda_j) : \forall (i,j), i \neq j$:

$$-D_j + \sum_{i \neq j} \frac{D_i}{B_i^j} \mu_{ij} + \theta - \lambda_j = 0 \quad \forall j \tag{7a}$$

$$\mu_{ij} \left(\frac{D_i}{B_i^j} r_j - (t - s) \right) = 0 \quad \forall (i,j) : i \neq j \tag{7b}$$

$$r_j \lambda_j = 0 \quad \forall j \tag{7c}$$

$$\mu_{ij} \geq 0 \quad \forall (i,j) : i \neq j \tag{7d}$$

$$\lambda_j \geq 0 \quad \forall j \tag{7e}$$

$$\frac{D_i}{B_i^j} r_j - (t - s) \leq 0 \quad \forall (i,j) : i \neq j \tag{7f}$$

$$\sum_j r_j - 1 = 0 \tag{7g}$$

$$r_j \geq 0 \quad \forall j \tag{7h}$$

where (7a) are the first stationary conditions, (7b) (7c) are the complimentary slackness conditions, (7d) (7e) are the dual feasibility conditions, and (7f) (7g) (7h) are the primal feasibility conditions.

Notice that for every site j, (7f) implies that there could be bottlenecking link if $\max_{i \neq j} \left\{ D_i / B_i^j \right\} > t - s$. There would not be a bottlenecking link only if making $r_j = 1$ meets the deadline. Let us denote

$$U_j := \max_{i \neq j} \left\{ D_i / B_i^j \right\} \tag{8}$$

DOI: 10.475/123_4

as the maximum upload time for site j if $r_j = 1$, and denote

$$b_j := \arg\max_{i \neq j} \left\{ D_i / B_i^j \right\} \tag{9}$$

as the set of bottlenecking data sources. If for any $i \notin b_j$, then (7f) is satisfied if it satisfied for b_j and also $D_i/B_i^j < t - s$. This fact and (7b) implies that $\mu_{ij} = 0$ if $i \notin b_j$. Now the KKT conditions (7) have redundant conditions that can be eliminated to:

$$\begin{aligned}
-D_j + U_j \sum_{i \in b_j} \mu_{ij} + \theta - \lambda_j &= 0 \quad \forall j && (10a)\\
\mu_{ij}\left(U_j r_j - (t-s)\right) &= 0 \quad \forall (i,j) : i \in b_j && (10b)\\
r_j \lambda_j &= 0 \quad \forall j && (10c)\\
\mu_{ij} &\geq 0 \quad \forall (i,j) : i \in b_j && (10d)\\
\lambda_j &\geq 0 \quad \forall j && (10e)\\
U_j r_j - (t-s) &\leq 0 \quad \forall (i,j) : i \in b_j && (10f)\\
\sum_j r_j - 1 &= 0 && (10g)\\
r_j &\geq 0 \quad \forall j && (10h)
\end{aligned}$$

We have three cases for each r_j:

(1) $r_j = (t-s)/U_j$. Since $t - s > 0$ and $U_j > 0$, then $r_j > 0$. Then (10c) results in $\lambda_j = 0$. In this case (10a) turns into:

$$\theta = D_j - U_j \sum_{i \in b_j} \mu_{ij} \tag{11}$$

Since $\sum_{i \in b_j} \mu_{ij} \geq 0$ from (10d), then this means that $\theta \leq D_j$.

(2) $0 < r_j < (t-s)/U_j$. Then (10c) results in $\lambda_j = 0$ and (10b) results in $\mu_{ij} = 0 : \forall i \in b_j$. In this case (10a) turns into:

$$\theta = D_j \tag{12}$$

(3) $r_j = 0$. Then (10b) results in $\mu_{ij} = 0 : \forall i \in b_j$. In this case (10a) turns into:

$$\theta = D_j + \lambda_j \tag{13}$$

Since (10e), then $\theta \geq D_j$.

Since θ can only be a single value, the above cases give a natural ordering. For a particular θ: if $D_j > \theta$, then Case 1 applies; if $D_j < \theta$ then Case 3 applies; if $D_j = \theta$ then Cases 1,2, or 3 could apply. This also means that we can restrict θ to the set $\left\{D_j : \forall j\right\}$ without restricting the solution space of the task placements $r_j : \forall j$.

Also, if all sites are in Case 1 and we observe that from (10g) $\sum_j \frac{1}{U_j} < \frac{1}{t-s}$, then the deadline t is too soon and the problem is infeasible. Let us denote the lower bound on t as:

$$\underline{t} := s + \frac{1}{\sum_j \frac{1}{U_j}} \tag{14}$$

On the other hand, let $l := \arg\max_j \{D_j\}$ and if $U_l < t - s$, then $r_l = 1$ and $r_j = 0 : \forall j \neq l$. Let us denote the upper bound on t as:

$$\bar{t} := s + U_l \tag{15}$$

If $t \geq \bar{t}$, then $r_l = 1$, $r_j = 0 : \forall j \neq l$, and the WAN usage is $\sum_i D_i - D_l$.

If $t = \underline{t}$, then $r_j = (\underline{t} - s)/U_j : \forall j$ and the WAN usage is $\sum_i D_i - (\underline{t} - s)\sum_j (D_j/U_j)$.

Note that if either the maximum deadline range

$$\bar{t} - \underline{t} = U_l - \frac{1}{\sum_j \frac{1}{U_j}} \tag{16}$$

$$\tag{17}$$

or the maximum WAN savings

$$D_l - \frac{1}{\sum_j \frac{1}{U_j}} \sum_j \frac{D_j}{U_j} \tag{18}$$

has significant cost, then developing an optimization algorithm is worthwhile.

We have the following algorithm:

(1) **Initialize:**
- Order D_j in descending order and relabel the indexes for this ordering.
- Set $y := 1$, $k := 1$, and $r_j := 0 : \forall j$
- For each site j, set:
  $U_j := \max_{i \neq j} \left\{D_i / B_i^j\right\}$
  $b_j := \arg\max_{i \neq j} \left\{D_i / B_i^j\right\}$.

(2) **Test for feasibility:**
- If $t \leq s + \frac{1}{\sum_j \frac{1}{U_j}}$, then stop because the problem is infeasible.

(3) **Process:**
- Replace $r_k := \min\{y, (t-s)/U_k\}$.
- Update $y := y - r_k$ and $k := k + 1$.
- If $y \leq 0$ or $k > |\mathcal{L}|$, then stop and output $r_j : \forall j$. Otherwise repeat Step (3).

APPENDIX