On Scheduling Coflows\textsuperscript{[1]}

Saba Ahmadi, Samir Khuller, Manish Purohit, Sheng Yang

\textsuperscript{[1]} Appeared in IPCO 2017
Communication is Crucial!

Facebook analytics jobs spend 33% of their runtime in communication\(^\text{[1]}\)!

\[\text{[1]}\text{ Chowdhury et al. Managing Data Transfers in Computer Clusters with Orchestra, SIGCOMM’2011}\]
Model[1]

Input ports | Output ports
---|---
I₁ | O₁
| 1 |
I₂ | O₂
| 2 |
I₃ | O₃
| 2 |

- m×m switch
- Coflows: Collection of Parallel flows for a Common Goal
  - Coflow j is presented by a matrix $D_j$
    \[
    D_j = \begin{bmatrix}
    1 & 0 & 0 \\
    1 & 2 & 1 \\
    0 & 2 & 2 \\
    \end{bmatrix}
    \]
  - Capacity Constraint 1 at all ports.
  - At any time slot, scheduled flows form a matching.
    \[
    L_{ij} = \text{deg}_{G_j}(i)
    \]

Model: Scheduling a Single Coflow

Application of Hall’s Theorem
Model: Scheduling Multiple Coflows

Weighted Completion
Time: 4*10+9*5=85

Weight = 10

Weight = 5

Time 1

Time 2

Time 3

Time 4

Time 5

Time 6

Time 7

Time 8

Time 9
Model: Scheduling Multiple Coflows

Weighted Completion Time: $5 \times 5 + 9 \times 10 = 115$
Model: Scheduling Multiple Coflows

Weighted Completion Time: \(4 \times 10 + 7 \times 5 = 75\)
Given $n$ coflows, find a feasible schedule which minimizes total weighted completion time, i.e. $\sum_{j=1}^{n} w_j C_j$.
Prior Work

Systems

• Chowdhury and Stoica (HOTNETS 2012), Coflow: A Networking Abstraction for Cluster Applications
• Chowdhury, Zhong and Stoica (SIGCOMM 2014), Efficient Coflow Scheduling with Varys
• Chowdhury and Stoica (SIGCOMM 2015), Effective Coflow Scheduling Without Prior Knowledge
## Prior Work

### Theory

<table>
<thead>
<tr>
<th></th>
<th>Deterministic, No release time</th>
<th>Deterministic, With release time</th>
<th>Randomized, No release time</th>
<th>Randomized, With release time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qiu, Stein and Zhong - SPAA15</td>
<td>64/3</td>
<td>67/3</td>
<td>15.5</td>
<td>16.5</td>
</tr>
<tr>
<td>Khuller and Purohit - SPAA16</td>
<td>8</td>
<td>12</td>
<td></td>
<td>$3 + \sqrt{2}$</td>
</tr>
<tr>
<td>This paper - IPCO17</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shafiee, Ghaderi – SPAA17</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview of our Approach

1. Solve an LP
2. Obtain an ordering of Coflows
3. Find feasible Coflow schedule
Step 1: Solve an LP

\[
L_{ij} = \deg_{G_j}(i)
\]

\[
M = I \cup O
\]

\[
\begin{align*}
\min \sum_{j \in J} w_j C_j \\
\text{subject to,} \quad C_j &\geq r_j + L_{i,j} \\
\sum_{j \in S} L_{i,j} C_j &\geq \frac{1}{2} \left( \sum_{j \in S} L_{i,j}^2 + \left( \sum_{j \in S} L_{i,j} \right)^2 \right) \\
\forall j \in J, &\forall i \in M \\
\forall i \in M, &\forall S \subseteq J
\end{align*}
\]

How can we get the second constraint?
Step 1: Solve an LP

\[
\sum_{j \in S} L_{i,j} C_j \geq \frac{1}{2} \left( \sum_{j \in S} L_{i,j}^2 + \left( \sum_{j \in S} L_{i,j} \right)^2 \right) \quad \forall S \subseteq J, \forall i \in M
\]

w.l.o.g assume:
\[
S = \{1, 2, \ldots, x\}
\]
\[
C_1 \leq C_2 \leq \cdots \leq C_x
\]

Step 1: Solve the LP

\[
\min \sum_{j \in J} w_j C_j
\]

subject to,

\[
C_j \geq r_j + L_{i,j} \quad \forall j \in J, \forall i \in M
\]

\[
\sum_{j \in S} L_{i,j} C_j \geq \frac{1}{2} \left( \sum_{j \in S} L_{i,j}^2 + \left( \sum_{j \in S} L_{i,j} \right)^2 \right) \quad \forall i \in M, \forall S \subseteq J
\]

The number of constraints is exponential, we can solve this LP using ellipsoid method.

Separation Oracle:
Given a solution \( C \), w.l.o.g assume \( C_1 \leq C_2 \cdots \leq C_n \). Let \( S_1 = \{1\} \), \( S_2 = \{1, 2\}, \ldots, S_n = \{1, 2, \cdots, n\} \). It’s sufficient to check whether the constraints are violated for the \( n \) sets \( S_1, \cdots, S_n \).

Not Practical!
Overview of our Approach

1. Solve an LP
2. Obtain an ordering of Coflows
3. Find feasible Coflow schedule
Step 3: Find a Feasible Coflow Schedule

Scheduling coflows sequentially:
Total weighted completion time = 100*1+200*1+300*1=600

Add an edge from a future coflow if it does not increase degree of the current coflow.

Total weighted Completion Time: 100*1+101*1+100*1 =301
Algorithm

Solve the LP to get the completion times of Coflows. Assume $C_1 \leq C_2 \cdots \leq C_n$.
Consider the Coflows in the same order: $G_1, \cdots, G_n$

for $j = 1, 2, \cdots, n$:
\[ G'_j = G_j \]
for $j = 1, 2, \cdots, n - 1$:
for $k = j + 1, j + 2, \cdots, n$:
Move $e \in G'_k$ to $G'_j$ if $\Delta(G'_j)$ does not increase.

Schedule new Coflows $G'_1, \cdots, G'_n$ using application of Hall’s Theorem.
Step 3: Find a Feasible Coflow Schedule

- Solve the LP to get the ordering $G_1, G_2, \ldots, G_n$
- Move edges backward to get $G'_1, G'_2, \ldots, G'_n$
- Schedule coflows $G'_1, G'_2, \ldots, G'_n$ using application of Hall’s theorem.

$c_1(alg) = \Delta(G'_1)$
$c_2(alg) \leq \Delta(G'_1) + \Delta(G'_2)$
$c_3(alg) \leq \Delta(G'_1) + \Delta(G'_2) + \Delta(G'_3)$
$c_j(alg) \leq \sum_{k \leq j} \Delta(G'_k)$
Cost of Algorithm

Lemma 1: For all $j \in \{1, 2, \cdots, n\}$, $\sum \Delta(G'_k) \leq 2\Delta(\bigcup_{k \leq j} G_k)$

\[ j = 2 : \]
\[ \sum_{k \leq j} \Delta(G'_k) = \Delta(G'_1) + \Delta(G'_2) = 2 + 2 = 4 \]
\[ \Delta(\bigcup_{k \leq j} G_k) = \Delta(G_1 \cup G_2) = 3 \]
\[ \rightarrow \sum_{k \leq j} \Delta(G'_k) \leq 2\Delta(\bigcup_{k \leq j} G_k) \]
Cost of Algorithm:

Lemma 1: For all \( j \in \{1, 2, \ldots, n\} \), \( \sum_{k \leq j} \Delta(G'_k) \leq 2 \Delta(\bigcup_{k \leq j} G_k) \)

Lemma 2: \( C_j \geq \frac{1}{2} \max_i \sum_{k=1}^{j} L_{i,k} = \frac{1}{2} \Delta(\bigcup_{k \leq j} G_k) \)

\( L_{ij} = deg_{G_j}(i) \)

Theorem 1:

\[
C_j(alg) = \sum_{k \leq j} \Delta(G'_k) \leq 2 \Delta(\bigcup_{k \leq j} G_k) \leq 4C_j
\]

\[
\sum_{j=1}^{n} w_j C_j(alg) \leq 4 \sum_{j=1}^{n} w_j C_j \leq 4OPT
\]

Theorem 2: There exists a deterministic combinatorial 5 approximation algorithm for coflow scheduling with release times.
Proof of Lemma 1:

Lemma 1: For all \( j \in \{1, 2, \cdots, n\} \),
\[
\sum_{k \leq j} \Delta(G'_{k}) \leq 2\Delta(\bigcup_{k \leq j} G_{k})
\]

Proof: For the simple case when \( j = n \) we’ll show:
\[
\sum_{k \leq n} \Delta(G'_{k}) \leq 2\Delta(\bigcup_{k \leq n} G_{k})
\]

After all edges are moved backward, consider a vertex of maximum degree in \( G'_n \).
Assume it’s \( u \).
\[\bar{G}_n = \bigcup_{k \leq n} G_k\]

\(G_1, \ldots, G_n: \) Initial Coflows

\(G'_1, \ldots, G'_n: \) Coflows after moving edges backward.

\(S_u \subseteq \{G'_1, G'_2, \ldots, G'_n\} \) is the set of modified coflows (after moving edges backward) where \(u\) has max degree.

\(S_v \subseteq \{G'_1, G'_2, \ldots, G'_n\} \) is the set of modified coflows where \(v\) has max degree.

\(S_u \cup S_v = \{G'_1, G'_2, \ldots, G'_n\}\)

\[\Delta(G_n) \geq \max\{\deg_{\bar{G}_n}(u), \deg_{\bar{G}_n}(v)\}\]

\[\deg_{\bar{G}_n}(u) \geq \sum_{G \in S_u} \deg_G(u)\]

\[\deg_{\bar{G}_n}(v) \geq \sum_{G \in S_v} \deg_G(v)\]

\[\Delta(\bar{G}_n) \geq \max\{\sum_{G \in S_u} \deg_G(u), \sum_{G \in S_v} \deg_G(v)\}\]

\[= \max\{\sum_{G \in S_u} \Delta(G), \sum_{G \in S_v} \Delta(G)\}\]

\[\sum_{k \leq n} \Delta(G'_k) = \sum_{G \in S_u} \Delta(G) + \sum_{G \in S_v} \Delta(G)\]

\[\leq 2(\max\{\sum_{G \in S_u} \Delta(G'), \sum_{G \in S_v} \Delta(G)\})\]

\[\sum_{k \leq n} \Delta(G'_k) \leq 2(\Delta(\bar{G}_n)) = 2(\Delta(\bigcup_{k \leq n} G_k))\]
Can we make it combinatorial?
Overview of Our Approach

Primal Dual Algorithm → Obtain an Ordering of Coflows → Find Feasible Coflow Schedule

Inspired by
Davis, Gandhi and Kothari [2013]
Mastrolilli, Queyranne, Schulz, Svensson and Uhan [2010]
Primal Dual Algorithm

**Primal LP:**

\[
\begin{align*}
\text{min} & \quad \sum_{j \in J} w_j C_j \\
\text{subject to,} & \quad C_j \geq r_j + L_{i,j} \\
& \quad \sum_{j \in S} L_{i,j} C_j \geq \frac{1}{2} \left( \sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2 \right) \quad \forall i \in M, \forall S \subseteq J
\end{align*}
\]

**Dual LP:**

\[
\begin{align*}
\text{max} & \quad \sum_{j \in J} \sum_{i \in M} \alpha_{i,j} (r_j + L_{i,j}) + \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} \frac{1}{2} \left( \sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2 \right) \\
\text{subject to,} & \quad \sum_{i \in M} \alpha_{i,j} + \sum_{i \in M} \sum_{S \subseteq J} L_{i,j} \beta_{i,S} \leq w_j \quad \forall j \in J \\
& \quad \alpha_{i,j} \geq 0 \quad \forall j \in J, i \in M \\
& \quad \beta_{i,S} \geq 0 \quad \forall i \in M, \forall S \subseteq J
\end{align*}
\]

Exponential number of variables
Polynomial number of constraints
Primal Dual Algorithm

\[
\max \sum_{j \in J} \sum_{i \in M} \alpha_{i,j} (r_j + L_{i,j}) + \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} \frac{1}{2} \left( \sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2 \right)
\]

\( J \) is the set of unscheduled jobs

Initialization:
\( J = \{1, 2, \cdots, n\} \)
\( \alpha_{i,j} = 0 \quad \forall i \in M, j \in J \)
\( \beta_{i,S} = 0 \quad \forall i \in M, S \subseteq J \)

\( k = n, n - 1, \cdots, 1 \)

\[
L_i = \sum_{j \in J} L_{ij}
\]

\[
\mu(k) = \arg\max_{i \in M} L_i
\]

\[
j = \arg\max_{\ell \in J} r_\ell
\]

Increase \( \beta_{\mu(k), j} \) till the constraint gets tight for a job \( j' \)

\[
\sigma(k) \leftarrow j'
\]

\[
r_j > \kappa \cdot L_{\mu(k)}
\]

False \quad True

\[
J \leftarrow J \setminus \sigma(k)
\]

\[
L_i \leftarrow L_i - L_{i, \sigma(k)}, \forall i \in M
\]

\( k \leftarrow \cdots \)

Output permutation \( \sigma(1), \sigma(2), \cdots, \sigma(n) \)
Overview of our Approach

1. Primal Dual Algorithm
2. Obtain an Ordering of Coflows
3. Find Feasible Coflow Schedule
Cost of Combinatorial Algorithm

Lemma 3: If there is an algorithm that generates a feasible coflow schedule such that for any coflow $j$, $C_j(alg) \leq a \max_{k \leq j} r_k + b\Delta(\bigcup_{k \leq j} G_k)$ for some constants $a$ and $b$, then the total cost of the schedule is bounded as follows.

$$
\sum_{j} w_j C_j(alg) \leq (a + \frac{b}{\kappa}) \sum_{j=1}^{n} \sum_{i \in M} \alpha_{i,j} r_j + 2(a\kappa + b) \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} f_i(S)
$$
**Cost of Combinatorial Algorithm**

\[ f_i(S) = \frac{\sum_{j \in S} L_{i,j}^2 + (\sum_{j \in S} L_{i,j})^2}{2} \]

**Lemma 3:** If there is an algorithm that generates a feasible coflow schedule such that for any coflow \( j \), \( C_j (\text{alg}) \leq a \max_{k \leq j} r_k + b \Delta (\bigcup_{k \leq j} G_k) \) for some constants \( a \) and \( b \), then the total cost of the schedule is bounded as follows.

\[
\sum_j w_j C_j (\text{alg}) \leq (a + \frac{b}{\kappa}) \sum_{j=1}^{n} \sum_{i \in M} \alpha_{i,j} r_j + 2(a\kappa + b) \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} f_i(S)
\]

---

**Primal Dual Algorithm**

Obtain an Ordering of Coflows

Find Feasible Coflow Schedule

\[ G_1, G_2, \cdots, G_n \]

**Lemma 1:** \( C_j (\text{alg}) \leq 2\Delta (\bigcup_{k \leq j} G_k) \)

\[ a = 0, b = 2 : \sum_{j=1}^{n} w_j C_j (\text{alg}) \leq \left( \frac{2}{\kappa} \right) \sum_{j=1}^{n} \sum_{i \in M} \alpha_{i,j} r_j + 2(2) \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} f_i(S) \]
Cost of Combinatorial Algorithm

$$\frac{2}{\kappa} = 2(2) \rightarrow \kappa = \frac{1}{2}$$

$$\sum_{j=1}^{n} w_j C_j(\text{alg}) \leq 4 \left( \sum_{j=1}^{n} \sum_{i \in M} \alpha_{i,j} r_j + \sum_{i \in M} \sum_{S \subseteq J} \beta_{i,S} f_i(S) \right) \leq 4OPT$$

5-approximation algorithm for coflow scheduling with release times!
4-approximation algorithm for coflow scheduling without release times!
3-approximation combinatorial algorithm for concurrent open shop with release times!
Coflow scheduling generalizes concurrent open shop.
Open Problems

• Considering flow time instead of completion time.

• Since coflow scheduling generalizes concurrent open shop, it is NP-hard to approximate it within a factor better than $(2-\varepsilon)^{[1]}$. We have an approximation factor of 4!
Thank You!