Robust Online Coflow Scheduling from Predictions

Olivier BRUN[1], Balakrishna J. PRABHU[1], Aminata SANGHO[1]

{brun, bala, sangho}@laas.fr

Keywords : *Sincronia, Approximation ratio, Online Scheduling.*

1 Introduction

Modern cloud platforms run large-scale parallel applications such as MapReduce, Hadoop, and Spark, where data transfers during communication stages frequently become the bottleneck for job completion. Following the introduction of coflows by Chowdhury and Stoica [1] in 2012, coflow scheduling has become the standard abstraction for coordinating these transfers. Existing work largely focuses on two extremes: the clairvoyant setting [2], where all flow sizes are known in advance, and the non-clairvoyant setting [3], where only port information is available as jobs are available in the system.

In this work, we examine a more realistic intermediate scenario where jobs arrive online and the scheduler receives predicted flow sizes. Since these predictions are inherently uncertain and may be unreliable, we investigate a fundamental question: how can a scheduler effectively leverage imperfect predictions to improve online coflow scheduling while providing provable performance guarantees?

2 Problem Statement and Main Results

Assuming that the fabric core can sustain 100% throughput and only the ingress and egress ports are congestion points, we model the datacenter as a non-blocking switch with m input and m output ports, each having unit capacity. Let $L = \{1, \dots, 2m\}$ denote the set of ports and $C = \{1, \dots, n\}$ the set of coflows. Each coflow k consists of a set of flows F_k, where each flow j is associated with a specific input–output port pair and has an unknown volume $v_{k,j} \in \mathbb{N}^*$. Coflow k arrives at time r_k with weight w_k. Our objective is to design an online scheduling algorithm that minimizes the weighted coflow completion time

$$\sum_{k \in C} w_k C_k,$$

where C_k denotes the completion time of coflow $k \in C$ (i.e., when its last flow finishes).

We assume that true flow sizes are unavailable, and the scheduler must rely on unreliable machine learning predictions. When coflow k arrives, the scheduler receives predicted flow sizes $\hat{v}_{k,j} \in \mathbb{N}^*$ satisfying

$$\frac{\hat{v}_{k,j}}{\mu_{\max}} \leq v_{k,j} \leq \frac{\hat{v}_{k,j}}{\mu_{\min}},$$

where $\mu_{\min}$ and $\mu_{\max}$ are known constants. We denote by $\nu = \frac{\mu_{\max}}{\mu_{\min}}$ the prediction error ratio. In the online setting, the scheduler has no advance information about coflow release dates or flow sizes, but does receive these bounded predictions upon arrival.

Our framework builds upon two key algorithmic components:

- Select-and-Permute (SP) [5] operates in epochs, each consisting of two steps. First, it selects a subset of released but unscheduled coflows using an (α, β)-approximation algorithm for the Minimum Unscheduled Weight Problem (MUWP). Then, it schedules and executes this subset using an offline permutation-based algorithm. In the clairvoyant setting, the authors proved this framework achieves an $8+\gamma$ competitive ratio with an additive $2W$ term compared to optimal scheduling, where $W = \sum_{k \in C} w_k$ is the total weight and γ the approximation guarantee of the offline algorithm.

- SINCRONIA [2] is an offline algorithm that computes a scheduling order through a primal-dual approach, then performs greedy allocation following this order. Brun et al. [4] studied this algorithm under predictions and established an approximation ratio of $\gamma = \min(4\nu^2, \frac{W}{w_{\min}})$, where $W = \sum_{k \in C} w_k$ and $w_{\min} = \min_k w_k$.

Our main contribution is adapting these existing approaches to derive provable competitive ratio guarantees for coflow scheduling with predictions. In the proposed scheme, SINCRONIA is used as the scheduling algorithm in the permute step of SP. More precisely, our proposed online coflow scheduling algorithm with predictions proceeds as follows:

Algorithm 1 Online Coflow Scheduling with Predictions

```
$k \leftarrow 0$, $\tau_0 \leftarrow 0$
while $\tau_k < T$ do
    $A_k = \{j \in J \mid r_j \le \tau_k\}$                ▷ Coflows released by time $\tau_k$
    $R_k = A_k \setminus \bigcup_{t<k} S_t$                ▷ Coflows released but not yet scheduled
    Select $S_k \subseteq R_k$ by solving MUWP using an $(\alpha, \beta)$-approximation
    Schedule coflows in $S_k$ using Sincronia with predicted sizes
    $k \leftarrow k+1$, $\tau_k \leftarrow 2^{k-1}$
end while
```

We establish that Algorithm 1 achieves a competitive ratio of $4\nu(4+\nu)$ with an additive term of $4\nu W$, where $W = \sum_{k \in C} w_k$ is the total weight. In the special case of perfect predictions, where $\mu_{\min} = \mu_{\max} = 1$ (and thus $\nu = 1$), our algorithm guarantees a competitive ratio of 20 against the optimal offline clairvoyant scheduler.

3 Conclusion

We developed a robust online framework that achieves a provable competitive ratio. Numerical experiments demonstrate that the framework maintains strong performance even when predictions are highly inaccurate.

References

[1] Mosharaf Chowdhury and Ion Stoica. Coflow: A networking abstraction for cluster applications. In *HotNets*, , pages 31–36, Redmond, Washington, 2012.

[2] Saksham Agarwal, Rachit Agarwal, Shijin Rajakrishnan, David Shmoys, Akshay Narayan, and Amin Vahdat. Sincronia: Near-Optimal Network Design for Coflows. In Proc. *ACM SIGCOMM*, pages 16–29, 2018.

[3] Akhil Bhimaraju, Debanuj Nayak, Rahul Vaze. Non-clairvoyant Scheduling of Coflows. *WiOpt*, 2020.

[4] Olivier Brun, Balakrishna Prabhu and Oumayma Haddaji. Prediction-based Coflow Scheduling. *HAL*, 2023.

[5] Samir Khuller, Jingling Li, Pascal Sturmfels, Kevin Sun, and Prayaag Venkat. Select and permute: An improved online framework for scheduling to minimize weighted completion time. *Theoretical Computer Science*, 795:420–431, 2019.