SOLACE Seminars

Prediction-based Coflow Scheduling in Datacenter Networks

Olivier Brun, Balakrishna J. Prabhu, Oumayma Haddaji

LAAS–CNRS, Toulouse, France

October 12^{th}, 2023

Outline

INTRODUCTION

PROBLEM FORMULATION AND EXISTING WORKS

COFLOW SCHEDULING WITH PREDICTIONS

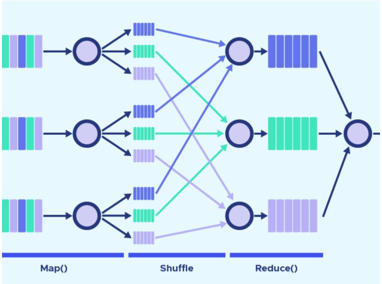
NUMERICAL RESULTS

CONCLUSION

INTRODUCTION

Context

- Distributed computing frameworks: Hadoop MapReduce, Apache Spark
- Massive data transfers in datacenter networks (e.g., shuffle phase)
  - For some workloads, they can account for more than 50% of job completion times



- Coflow: set of concurrent flows related to a common task

Coflow scheduling

- Minimization of average Coflow Completion Time (CCT)
- Clairvoyant setting
  - ✔ Source and destination ports as well as the precise volume of each flow are revealed upon the arrival of a coflow.
  - ✔ NP-hard, inapproximable below a factor 2
  - ✔ Efficient approximation algorithms, e.g., Varys or Sincronia[1]
- Non-clairvoyant setting
  - ✔ Flow sizes remains unknown
  - ✔ Scheduling schemes generalizing the *LAS* (e.g., Aalo) or *RR* (e.g., BlindFlow) scheduling disciplines.

1

- ☞ M. Shafiee et *al.*, An improved bound for minimizing the total weighted completion time of coflows in datacenters, IEEE/ACM Trans. Netw., vol. 26, no. 4, 2018.
- ☞ S. Agarwal et *al.*, Sincronia: Near-optimal network design for coflows. in Proc. ACM SIGCOMM, 2018.
- ☞ M. Chowdhury et *al.*,. Near optimal coflow scheduling in networks, in Proc. ACM SPAA, 2019.

Contributions

- ML predictions are revealed to the coflow scheduler
  - ✔ Actual flow sizes remain unknown and predictions are unreliable
  - ✔ How to exploit predictions for coflow scheduling? Is it even advisable to do so?
- Approximation ratio of Sincronia as a function of the prediction error
- A Consistent and robust prediction-based coflow scheduling algorithm.

PROBLEM FORMULATION AND EXISTING WORKS

System model and notations

- Big-Switch model: capacity b_ℓ for port ℓ.
- Offline setting.
- Set $\mathcal{C} = \{1, 2, \ldots, n\}$ of coflows
  - ✔ Coflow k is a collection F_k of flows, where flow j has size $v^{k,j}$
  - ✔ $F_{k,\ell}$ is the set of flows of coflow k which use port ℓ
  - ✔ C_k denotes the CCT of coflow k
- Problem formulation

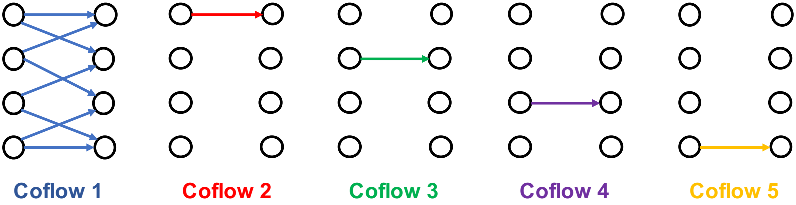
$$\min_{r} \sum_{k \in \mathcal{C}} C_k \tag{P1}$$

$$\text{s.t.} \sum_{k \in \mathcal{C}} \sum_{j \in F_{k,\ell}} r^{k,j}(t) \leq b_\ell, \quad \forall \ell \in \mathcal{L}, \forall t \in \mathcal{T}, \tag{1}$$

$$\int_0^{C_k} r^{k,j}(t)\, dt \geq v^{k,j}, \quad \forall j \in F_k, \forall k \in \mathcal{C}, \tag{2}$$

Example

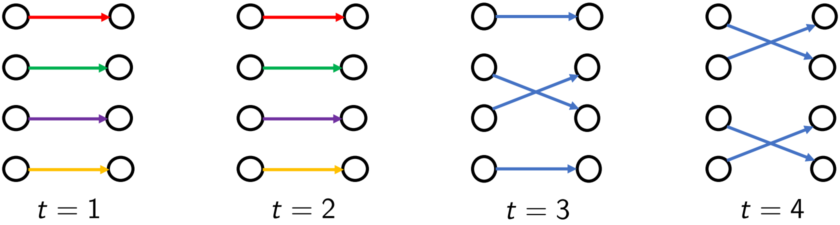
- All fabric ports have the same normalized bandwidth of 1
- All flows of coflow 1 have volume 1
- All other flows have volume $2 + \epsilon$



- The goal is to allocate flow rates so as to minimize $(C_1 + C_2 + C_3 + C_4 + C_5)/5$.

Example – Clairvoyant offline optimum

- Time-indexed MILP formulation for the clairvoyant setting[2]



- Average CCT is $OPT = (4 + 4 \times 2)/5 = 2.4$

2

☞ Y. Magnouche et *al.*, Branch-and-benders-cut algorithm for the weighted coflow completion time minimization problem, INOC 2022.

Non-clairvoyant coflow scheduling – BlindFlow

- Round Robin allocation on port ℓ: $r_\ell(t) = b_\ell / n_\ell(t)$
- Generalized RR allocation:

$$r^{k,j}(t) = \min\left\{r_i(t), r_o(t)\right\} = \frac{1}{\max\left\{1/r_i(t), 1/r_o(t)\right\}}$$

  for ongoing flow $j \in F_k$ with ingress/egress ports i and o.

- BlindFlow rate allocation[3] : $r^{k,j}(t) = \frac{1}{1/r_i(t)+1/r_o(t)}$

Theorem

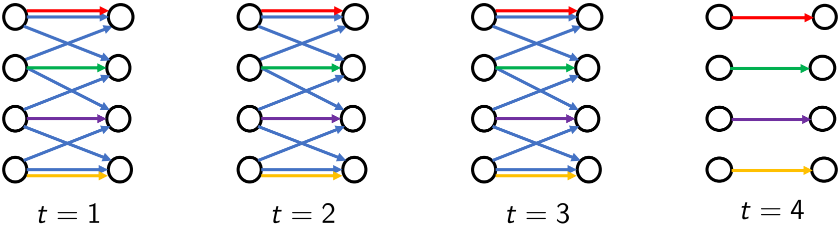
The rate allocation of BlindFlow is feasible and $8 \times p$ approximate, where $p = \max_{k \in \mathcal{C}} |F_k|$ is the maximum number of flows that any coflow can have.

3

☞ A. Bhimaraju, D. Nayak and R. Vaze, Non-clairvoyant scheduling of coflows, WiOpt 2020, 2020.

Example – Generalized RR allocation

- All fabric ports have the same normalized bandwidth of 1
- Flows of coflow 1 have volume 1, all others have volume 2



- Average CCT is $(3 + 4 \times 4)/5 = 3.8 \approx 1.6 \times OPT$ $(8 \times p = 64)$

Clairvoyant coflow scheduling – Sincronia

- Transport layer may not be able to enforce an arbitrary per-flow rate allocation.
- Sincronia orders the coflows in some appropriate order, and leverage priority forwarding mechanisms
  1. σ-order: coflow $\sigma(n)$ has priority over coflow $\sigma(n+1)$
  2. Greedy rate allocation: a flow is blocked iff ingress/egress port is busy serving a higher-priority flow

Clairvoyant coflow scheduling – Sincronia σ-order

- CCT of coflow k at port ℓ in isolation: $p_{\ell,k} = \sum_{j \in F_{k,\ell}} v_{k,j}/b_\ell$
- Method for computing the σ-order:

$$\text{Min} \sum_{k \in \mathcal{C}} C_k \qquad \text{(P3-Primal)}$$

s.t

$$\sum_{k \in S} p_{\ell,k} C_k \geq f_\ell(S), \;\; \ell \in \mathcal{L}, S \subseteq \mathcal{C},$$

$$C_k \geq 0, \;\; k \in \mathcal{C},$$

$$\text{Max} \sum_{\ell \in \mathcal{L}} \sum_{S \subseteq \mathcal{C}} f_\ell(S)\, y_{\ell,S} \qquad \text{(P3-Dual)}$$

s.t

$$\sum_{S: k \in S} \sum_{\ell \in \mathcal{L}} p_{\ell,k} y_{\ell,S} \leq 1, \;\; k \in \mathcal{C},$$

$$y_{\ell,S} \geq 0, \;\; \ell \in \mathcal{L}, S \subseteq \mathcal{C}.$$

where $f_\ell(S) = \frac{1}{2} \sum_{k \in S} (p_{\ell,k})^2 + \frac{1}{2} \left(\sum_{k \in S} p_{\ell,k}\right)^2$.

- Problem P3-Primal is a relaxation of the original coflow scheduling problem

Clairvoyant coflow scheduling – Sincronia σ-order

- Sincronia primal-dual algorithm

Initialize all dual variables $y_{\ell,S}$ to 0 and set $w_k = 1$ for all $k \in \mathcal{C}$
$S \leftarrow \mathcal{C}$
for $t = n \dots 1$ **do**
  $b \leftarrow \text{argmax}_{\ell \in \mathcal{L}} \sum_{k \in S} p_{\ell,k}$ ▷ Bottleneck port
  $k^* \leftarrow \text{argmin}_{k \in S} \left(\frac{w_k}{p_{b,k}} \right)$ ▷ Coflow with largest weighted proc. time
  $C_{k^*} \leftarrow \sum_{k \in S} p_{b,k}$ and $y_{b,S} \leftarrow \frac{w_{k^*}}{p_{b,k^*}}$ ▷ Set primal and dual variables
  $w_k \leftarrow w_k - w_{k^*} \frac{p_{b,k}}{p_{b,k^*}}$ for all $k \in S$ ▷ Update coflow weights
  $\sigma(t) \leftarrow k^*$ ▷ Set priority of coflow k^*
  $S \leftarrow S \setminus \{k^*\}$ ▷ Remove k^* from the set of unscheduled coflows
end for

Theorem

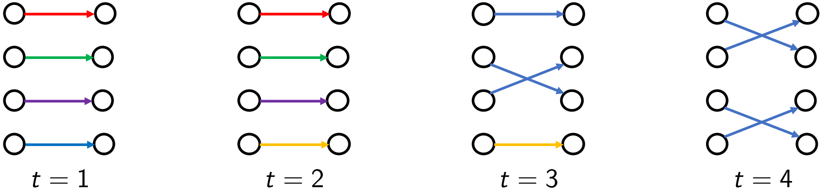
Sincronia provides a feasible solution to problem P3-Primal whose cost is at most $2\times$ the optimal cost. As the Greedy rate allocation is 2-optimal, Sincronia achieves an average CCT within $4\times$ of the optimal one.

Example – Sincronia

- σ-order

t	b	$\sigma(t)$	$\{w_1, w_2, w_3, w_4, w_5\}$	S
–	–	–	$\{1, 1, 1, 1, 1\}$	$\{1, 2, 3, 4, 5\}$
5	4	5	$\{\epsilon/(2+\epsilon), 1, 1, 1, 0\}$	$\{1, 2, 3, 4\}$
4	3	1	$\{0, 1, 1, 1-\epsilon/2, 0\}$	$\{2, 3, 4\}$
3	3	4	$\{0, 1, 1, 0, 0\}$	$\{2, 3\}$
2	2	3	$\{0, 1, 0, 0, 0\}$	$\{2\}$
1	1	2	$\{0, 0, 0, 0, 0\}$	$\emptyset$

- Greedy rate allocation with $\sigma = \{2, 3, 4, 1, 5\}$



- Average CCT is $(4 + 3 \times 2 + 3)/5 = 2.6 \approx 1.08 \times OPT$

COFLOW SCHEDULING WITH PREDICTIONS

Sincronia with predictions

- Sincronia is ran with predictions $\hat{v}^{k,j} = v^{k,j} + \Delta v^{k,j}$, where $\Delta v^{k,j}$ represents the prediction error
- Predicted transmission time of coflow $k \in \mathcal{C}$ on port $\ell \in \mathcal{L}$

$$\hat{p}_{\ell,k} = \sum_{j \in F_{k,\ell}} \frac{\hat{v}^{k,j}}{b_\ell} = p_{\ell,k} + \eta_{\ell,k},$$

- With $\mu_{min} = \min_{\ell,k} \left(\frac{\hat{p}_{\ell,k}}{p_{\ell,k}} \right)$ and $\mu_{max} = \max_{\ell,k} \left(\frac{\hat{p}_{\ell,k}}{p_{\ell,k}} \right)$,

$$\mu_{min}\, p_{\ell,k} \le \hat{p}_{\ell,k} \le \mu_{max}\, p_{\ell,k}, \qquad \text{for all } \ell \text{ and } k.$$

Sincronia with predictions

Theorem

Scheduling coflows in the order determined by Sincronia with predictions as inputs yields an average CCT which is at most $\min\left\{4 \times \left(\frac{\mu_{max}}{\mu_{min}}\right)^2, 2n\right\}$ *the optimal one.*

- The first upper bound depends on the prediction error, but the second one not (robustness).
- **Example**: if the relative prediction error on flow sizes is at most 50%, then $\mu_{min} \geq \frac{1}{2}$ and $\mu_{max} \leq \frac{3}{2}$, so that the performance guarantee is $\min\{36, 2n\}$.

A consistent and robust prediction-based algorithm

- Run Sincronia and RR in parallel
  - Sincronia uses predictions to schedule coflows in the fabric over a fraction λ of time,
  - RR schedules the coflows the rest of the time
  - The resulting rate allocation is

$$r^{k,j}(t) = \lambda \times r^{k,j}_{SP}(t) + (1-\lambda) \times r^{k,j}_{RR}(t)$$

Theorem

Running in parallel Sincronia with predictions and RR yields an algorithm with competitive ratio $\min\left(\frac{4}{\lambda}\left(\frac{\mu_{max}}{\mu_{min}}\right)^2,\ \frac{2}{\lambda}n,\ \frac{8p}{1-\lambda}\right)$

- The algorithm is $\min\left\{\frac{2}{\lambda}n, \frac{8p}{1-\lambda}\right\}$-robust and $\frac{4}{\lambda}$-consistent

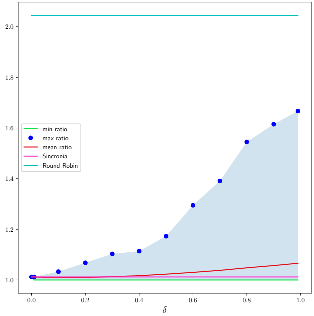
NUMERICAL RESULTS

Random Instances

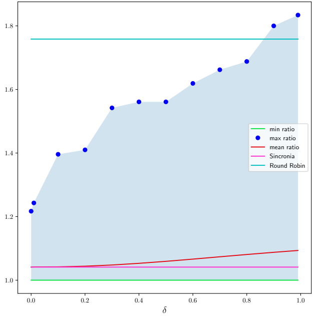
- Random instance generation
  - Number of coflows, number of ports and probability of a flow between two ingress/egress ports are given as inputs.
  - Flow volumes follow a (truncated) Gaussian distribution.
- Predictions
  - $\hat{v}^{k,j} = u^{k,j} \times v^{k,j}$ where $u^{k,j} \overset{iid}{\sim} U[1-\delta, 1+\delta]$.
  - $10,000$ predictions for each instance and each value of $\delta \in \{0, 0.01, 0.1, \ldots, 0.9, 0.99\}$.

Comparison against the clairvoyant optimum

- Instances with 6 coflows and 6 ports (10,000 predictions)



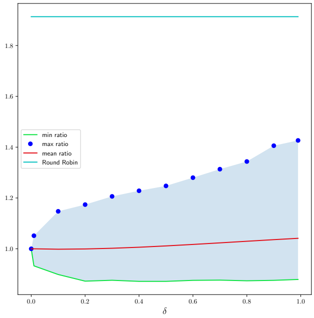
(a) One instance



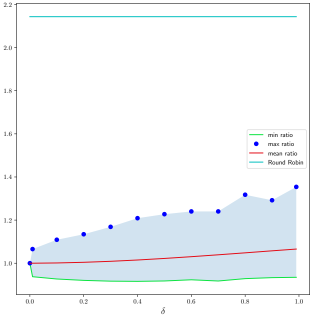
(b) 1,000 instances

Comparison against the clairvoyant Sincronia

- 100 instances with 10 ports and 15 or 30 coflows ($10,000$ predictions)



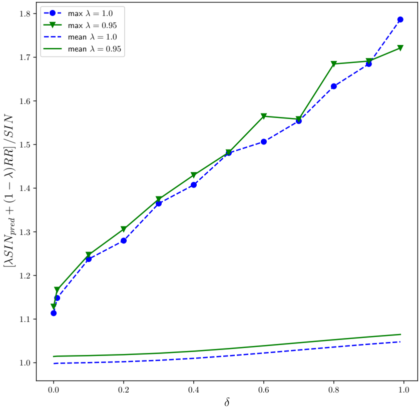
(a) 15 coflows



(b) 30 coflows

Combining Sincronia with predictions and RR

- 200 instances with 6 ports and 6 coflows $(20,000$ predictions)



Max and average values of $\frac{\lambda SIN_{pred}+(1-\lambda)RR}{SIN}$ for $\lambda = 0.95$ and $\lambda = 1.0$

CONCLUSION

Conclusion

- Coflow scheduling with unreliable predictions on flow sizes
- Sincronia with predictions as inputs
  - ✔ Approximation ratio
  - ✔ Sincronia performs well even when feed with terrible predictions
- No clear benefits in combining Sincronia with predictions and a RR rate allocation
- Operating Sincronia with ML predictions could be an efficient solution in practical scenarios

Questions?