Optimal Job Scheduling and Bandwidth Augmentation in Hybrid Data Center Networks

Binquan Guo, Zhou Zhang, Ye Yan, Hongyan Li

Xidian University, Xi′an, China

bqguo@stu.xidian.edu.cn

Preprint version: https://arxiv.org/abs/2209.11485

Dec.07 2022

Outline

1. Introduction
2. Related Work
3. System Model and Problem Formulation
4. Proposed Optimal Scheduling Scheme
5. Simulation Results
6. Conclusion

Introduction

Motivation

- Motivation: Data transfers account for $> 50\%$ of the job completion times in job computing systems, e.g., MapReduce, Pregel and Spark, becoming a bottleneck in data center networks (DCN).
- One key problem: Traditional fixed link capacity provision scheme can not fit the burst and dynamic data transfer patterns during a jobs execution and will slow down the job execution.

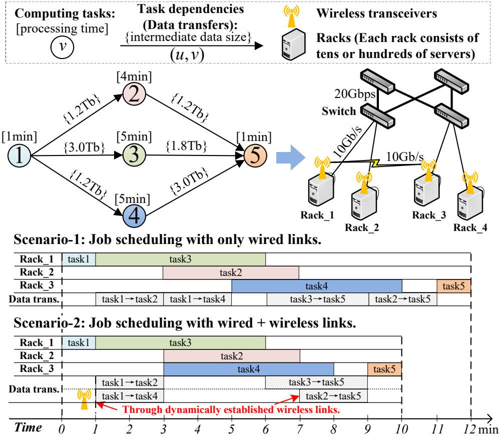
Adopting Wireless technology in DCNs

- Physical technologies: e.g., 60GHz mm Wave, Free space optics.
- Advantages: low switching latency, high throughput, low energy consumption.
- Challenges: Task dependency constraints, coupled constraints of computing and communication. + New constraints: Transceivers steering requirement, wireless channel interference.

Introduction

A motivated example

1. **Benefit:** By using dynamically established wireless links to transmit data, up to 16% of the job completion time can be reduced.
2. **Challenge:** Joint job scheduling and wireless bandwidth augmentation requires the scheduler and transceivers to make joint decisions under a lot of coupling constraints.



Computing tasks: [processing time] v
Task dependencies (Data transfers): {intermediate data size} (u,v)
Wireless transceivers
Racks (Each rack consists of tens or hundreds of servers)
[1min] 1
[4min] 2
[5min] 3
[5min] 4
[1min] 5
{1.2Tb}
{3.0Tb}
{1.2Tb}
{1.2Tb}
{1.8Tb}
{3.0Tb}
20Gbps
Switch
10Gb/s
10Gb/s
Rack_1
Rack_2
Rack_3
Rack_4
Scenario-1: Job scheduling with only wired links.
Rack_1 task1 task3
Rack_2 task2
Rack_3 task4 task5
Data trans. task1→task2 task1→task4 task3→task5 task2→task5
Scenario-2: Job scheduling with wired + wireless links.
Rack_1 task1 task3
Rack_2 task2
Rack_3 task4 task5
Data trans. task1→task2 task1→task4 task3→task5 task2→task5
Through dynamically established wireless links.
Time 0 1 2 3 4 5 6 7 8 9 10 11 12 min

Figure: An example to illustrate the advantages and challenges of using dynamically established wireless links to reduce job completion time.

Related work

Category 1: Flow Scheduling in Hybrid DCNs (no computing tasks)

9 M. Luo, et.al, Energy-efficient flow routing and scheduling in hybrid data center networks, IEEE GLOBECOM, 2019. (Our previous work.)

10 K. Han, et. al, Rush: Routing and scheduling for hybrid data center networks, in Proc. IEEE INFOCOM, 2015.

11 D. Halperin, et. al, Augmenting data center networks with multi-Gb wireless links, ACM SIGCOMM Comput. Commun. Rev., 2011.

12 Y. Cui, et. al, Dynamic scheduling for wireless data center networks, IEEE Trans. Parallel Distrib. Syst., 2013.

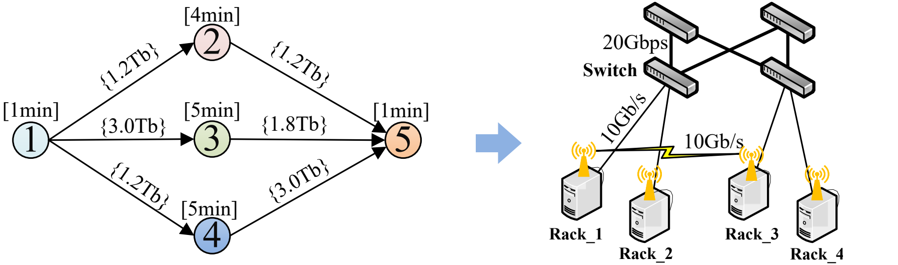
13 T. Li and S. Santini, Energy-aware coflow and antenna scheduling for hybrid server-centric data center networks, IEEE Trans. Green Commun. Netw., 2019.

Category 2: Joint task & flow scheduling in Hybrid DCNs

14 W. C. Ao, et. al, Joint workload distribution and capacity augmentation in hybrid DCNs, IEEE/ACM Trans. Netw., 2021. (The most related work to ours, ignoring task precedence constraints.))

System Model and Problem Formulation

1. A hybrid DCN consists of a set of racks $\mathcal{M} = \{1, 2, ..., M\}$. Each rack has wired links and wireless transceivers. The wireless bandwidth is divided into orthogonal subchannels set $\mathcal{K} = \{1, 2, ..., k, ...\}$ (shared among racks via FDMA), and each $k \in \mathcal{K}$ has a bandwidth B.
2. Each job is a directed acyclic graph $G = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is computing tasks set, and $\mathcal{E}$ is the data dependencies set. Each task $v \in \mathcal{V}$ requires certain computing resources with the processing time p_v. Each edge $(u, v) \in \mathcal{E}$ specifies the data size $d_{(u,v)}$ and its required bandwidth B_s.



System Model and Problem Formulation

Decision variables

- $x_{vi} = 1$: if computing task v is assigned to rack i, otherwise 0.
- s_v: The computing task v's start time.
- $y_{(u,v),k} = 1$: if data on edge (u, v) is transferred via subchannel k.
- $s_{(u,v)} = 1$: The start time of data transfer from task u to task v.

Parameters

- p_v: The computing task v's processing time.
- $q_{(u,v)}$: The transferring time of the data on edge (u, v) via wired links.
- $r_{(u,v)}$: The delay if data on edge (u, v) is transferred within a rack.
- $\tilde{q}_{(u,v)}$: The transferring time of the data from u to v via wireless subchannels.

Objective function

- The objective is to minimize the job completion time, which is

$$\min \ \max\{s_v + p_v \mid \forall v \in \mathcal{V}\}$$

System Model and Problem Formulation

A. Common Constraints for Computing Task Assignment

- 1. Non-repetition constraints: $\sum_{i \in \mathcal{M}} x_{vi} = 1, \forall v \in \mathcal{V}$.
- 2. Non-preemption constraints:
  $s_v + p_v \leq s_{v'}$ *or* $s_{v'} + p_{v'} \leq s_v$, *if* $\sum_{i \in \mathcal{M}} ix_{vi} = \sum_{i \in \mathcal{M}} ix_{v'i}$.
- 3. Precedence constraints: $s_u + p_u \leq s_v, \forall (u,v) \in \mathcal{E}$.

B. Constraints for Intermediate Data Transfers

- Definition: $z_{(u,v)} := 0$, *if* $\sum_{i \in \mathcal{M}} ix_{ui} = \sum_{i \in \mathcal{M}} ix_{vi}, \forall (u,v) \in \mathcal{E}$,
  $s_u + p_u + r_{(u,v)} \leq s_v, \forall (u,v) \in \mathcal{E}$, *if* $z_{(u,v)} = 0$.
- 4. Data transmitted through wired links
  $s_{(u,v)} + q_{(u,v)} \leq s_v, \forall (u,v) \in \mathcal{E}$, *if* $z_{(u,v)} = \alpha_{(u,v)} = 1$.
  $s_{(u,v)} + q_{(u,v)} \leq s_{(u',v')}$ or $s_{(u',v')} + q_{(u',v')} \leq s_{(u,v)}$.
- 5. Data transmitted via wireless subchannels
  $s_{(u,v)} + \check{q}_{(u,v)} \leq s_v$, *if* $z_{(u,v)} = 1$ *and* $\alpha_{(u,v)} = 0$.

$$s_{(u,v)} + q_{(u,v)} \leq s_{(u',v')} \text{ or } s_{(u',v')} + q_{(u',v')} \leq s_{(u,v)},$$

$$\textit{if} \sum_{k \in \mathcal{K}} ky_{(u,v),k} = \sum_{k \in \mathcal{K}} ky_{(u',v'),k} \textit{ and } \alpha_{(u,v)} = 0.$$

System Model and Problem Formulation

A. Common Constraints for Computing Task Assignment

- 1. Non-repetition constraints: $\sum_{i\in\mathcal{M}} x_{vi} = 1, \forall v \in \mathcal{V}$.
- 2. Non-preemption constraints: disjunctive (red) + logical constraints (blue)

  $s_v + p_v \leq s_{v'}$ *or* $s_{v'} + p_{v'} \leq s_v$, *if* $\sum_{i\in\mathcal{M}} i x_{vi} = \sum_{i\in\mathcal{M}} i x_{v'i}$.
- 3. Precedence constraints: $s_u + p_u \leq s_v, \forall (u,v) \in \mathcal{E}$.

B. Constraints for Intermediate Data Transfers

- Definition: $z_{(u,v)} := 0$, *if* $\sum_{i\in\mathcal{M}} i x_{ui} = \sum_{i\in\mathcal{M}} i x_{vi}, \forall (u,v) \in \mathcal{E}$,

  $s_u + p_u + r_{(u,v)} \leq s_v, \forall (u,v) \in \mathcal{E}$. *if* $z_{(u,v)} = 0$. logical constraints (blue)
- 4. Data transmitted through wired links:

  $s_{(u,v)} + q_{(u,v)} \leq s_v, \forall (u,v) \in \mathcal{E}$, *if* $z_{(u,v)} = \alpha_{(u,v)} = 1$.

  $s_{(u,v)} + q_{(u,v)} \leq s_{(u',v')}$ or $s_{(u',v')} + q_{(u',v')} \leq s_{(u,v)}$. disjunctive const.(red)
- 5. Data transmitted via wireless subchannels:

  $s_{(u,v)} + \check{q}_{(u,v)} \leq s_v$, *if* $z_{(u,v)} = 1$ *and* $\alpha_{(u,v)} = 0$.

  $s_{(u,v)} + q_{(u,v)} \leq s_{(u',v')}$ or $s_{(u',v')} + q_{(u',v')} \leq s_{(u,v)}$, disjunct.+logical const.

  $$\text{if} \sum_{k\in\mathcal{K}} k y_{(u,v),k} = \sum_{k\in\mathcal{K}} k y_{(u',v'),k} \text{ and } \alpha_{(u,v)} = 0.$$

System Model and Problem Formulation

Problem formulation

$$\textbf{OP}: \min_{\mathbf{s},\mathbf{x},\mathbf{y}} \ \max\{s_v + p_v \mid \forall v \in \mathcal{V}\}$$
$$\text{s.t. } (1) - (10),$$

where $\mathbf{s} = \{s_v, \forall v \in \mathcal{V}\} \cup \{s_{(u,v)}, \forall (u,v) \in \mathcal{E}\}$, $\mathbf{x} = \{x_{vi}, \forall v \in \mathcal{V}, \forall i \in \mathcal{M}\}$ and $\mathbf{y} = \{y_{(u,v),k}, \forall (u,v) \in \mathcal{E}, \forall k \in \mathcal{K}\}$.

Analysis and Observation

- **OP** is a complex Mixed Integer Non-linear Programming (MINLP), which is not directly solvable by existing optimization methods.
- The exhaustive search for the optimal solution is intractable, due to the huge solution space imposed by logical and disjunctive constraints.
- Even for a common scale **OP** (e.g., **job size** ≤ 10 in production cases), searching for the optimal solution is non-trivial, and the time complexity is unacceptable.

Proposed Optimal Scheduling Scheme

We transform OP into an equivalent problem based on combination of multiple steps, which paves the way for adopting the sophisticated optimization methods to acquire its optimal solution efficiently.

- **Fundamental bounds estimation**: Lower bound and upper bound.
- **Generalized data transfer model**: Depending on the assignment decisions of adjacent tasks, the intermediate data on each edge between adjacent tasks is either available in local disks, or transferred through wired or wireless links. Such conditional cases is the main cause of the non-linearity of the constraints in OP.

  Note: if adjacent task u and v are assigned to the same rack, the delay of transferring the data locally is denoted as $r_{(u,v)}$.
- **Constraints Decoupling and Linearization**: With the bounds and the generalized data transfer model, we can linearize OP based on the disjunction reformulation technique.
- **Decomposition and Acceleration**:To further speed up the solving procedure of using the Branch and Bound algorithm.

Proposed Optimal Scheduling Scheme

Upper bound

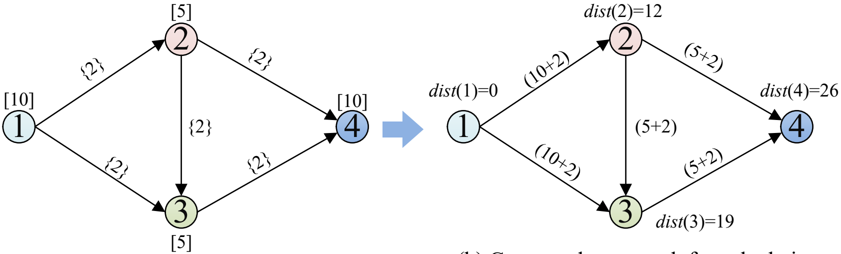
For any given job, we take $T_{max} = \sum_{v \in \mathcal{V}} p_v + \sum_{(u,v) \in \mathcal{E}} r_{(u,v)}$ as the job's upper bound by assuming all its tasks are assigned to a single rack.

Lower bound: The Longest Branch Algorithm

```
Define $c_{(u,v)}$ as the cost of edge $(u, v) \in \mathcal{E}$.
**for** each task $v \in \mathcal{V}$ **do**
    Initialize $dist(v) = 0$ as the distance from start to $v$.
    **for** each outgoing edge $(v, x)$ of task $v$ **do**
        Set $c_{(v,x)} = p_v + r_{(v,x)}$.
Topologically sort $\mathcal{V}$ in $G$.
**for** each task $v \in \mathcal{V}$ in topological sort order **do**
    Update $dist(v) = \max_{(u,v) \in \mathcal{E}} \{dist(v) + c_{(u,v)}\}$.
**return** $\max_{v \in \mathcal{V}} \{dist(v) + p_v\}$.
```

Proposed Optimal Scheduling Scheme

An example to show the lower bound estimation procedure.



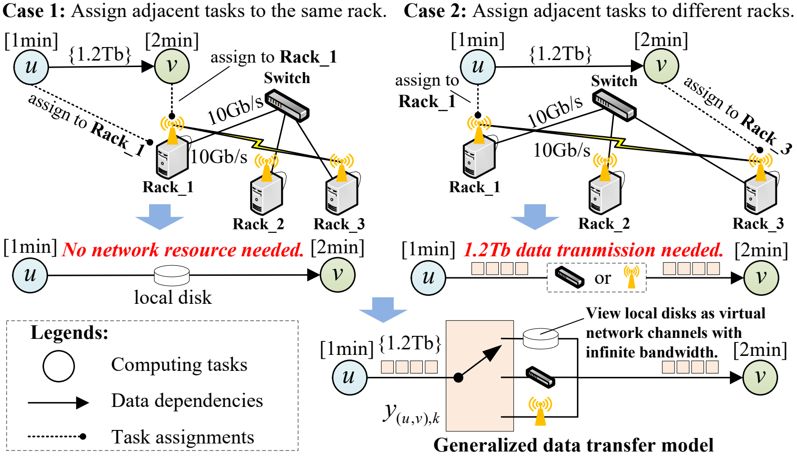
(a) An example DAG job graph

(b) Converted cost graph for calculating the longest branch

Figure: An example for calculating the longest branch of DAG job graph.

Proposed Optimal Scheduling Scheme

Technique-2: Generalized data transfer model.



Case 1: Assign adjacent tasks to the same rack.
Case 2: Assign adjacent tasks to different racks.
[1min]
{1.2Tb}
[2min]
u
v
assign to Rack_1
Switch
10Gb/s
10Gb/s
assign to Rack_1
Rack_1
Rack_2
Rack_3
[1min]
{1.2Tb}
[2min]
assign to Rack_1
Switch
10Gb/s
10Gb/s
assign to Rack_3
Rack_1
Rack_2
Rack_3
[1min] No network resource needed. [2min]
local disk
[1min] 1.2Tb data tranmission needed. [2min]
or
Legends:
Computing tasks
Data dependencies
Task assignments
[1min] {1.2Tb}
$y_{(u,v),k}$
View local disks as virtual network channels with infinite bandwidth.
[2min]
Generalized data transfer model

Figure: Illustration of generalized data transfer model.

Proposed Optimal Scheduling Scheme

Technique-3: Constraints Linearization.

A. Auxiliary variables definition

- $\tilde{x}_{vi} \in [0, T_{max}]$: $\tilde{x}_{vi} = \tau$ denotes task v is assigned to rack i and begins to process at time τ, otherwise $\tilde{x}_{vi} = 0$.
- $\tilde{y}_{(u,v),k} \in [0, T_{max}]$: $\tilde{y}_{(u,v),k} = \tau$ denotes that the intermediate data on edge (u, v) is assigned to channel $k \in \{b, c\} \cup \mathcal{K}$ and begins to transfer at time τ, otherwise $\tilde{y}_{(u,v),k} = 0$.
- $\psi_{vv'} \in \{0, 1\}^{|\mathcal{M}|}$: Task assignment indicator. $\psi_{vv'i} = 1$ indicates that task v and v' are assigned to the same rack i, otherwise 0.
- $\sigma_{vv'} \in \{0, 1\}$: Task precedence indicator. If task v starts no later than v', $\sigma_{vv'} = 1$; otherwise $\sigma_{vv'} = 0$.
- $\chi_{ee'} \in \{0, 1\}^{|\{b\} \cup \mathcal{K}|}$: Contention indicator. $\chi_{ee'k} = 1$ if the data on edge e and e' compete for the network channel k, otherwise 0.
- $\phi_{ee'} \in \{0, 1\}$: Flow precedence indicator. If the data on e begins to transfer no later than the data on e', $\phi_{ee'} = 1$, where $e, e' \in \mathcal{E}, e \neq e'$.

Next, some constraints are needed to construct auxiliary variables.

Proposed Optimal Scheduling Scheme

Technique-3: Constraints Decoupling and Linearization.

B. Auxiliary variables construction

- 1. Auxiliary variables for binding variables:
  $\tilde{x}_{vi} - 1 \leq x_{vi} \cdot T_{max} - (1 - x_{vi}) \cdot \varepsilon, \forall i \in \mathcal{M}.$
  $\tilde{y}_{ek} - 1 \leq y_{ek} \cdot T_{max} - (1 - y_{ek}) \cdot \varepsilon, \forall k \in \mathcal{K} \cup \{b, c\}, \varepsilon \in (0, 1).$
- 2. Auxiliary variables for precedence constraints:
  $\sum_{i \in \mathcal{M}} \psi_{vv'i} \leq 1, \forall v, v' \in \mathcal{V}, v \neq v'.$
  $\sum_{k \in \mathcal{K} \cup \{b\}} \chi_{ee'k} \leq 1, \forall e, e' \in \mathcal{E}, e \neq e'.$
  $0 \leq x_{vi} + x_{v'i} - 2 \cdot \psi_{vv'i} \leq 1, \forall i \in \mathcal{M}.$
  $0 \leq y_{ek} + y_{e'k} - 2 \cdot \chi_{ee'k} \leq 1, \forall k \in \mathcal{K} \cup \{b\}.$

C. Computing resource constraint reformulation

- $\sum_{i \in \mathcal{M}} \tilde{x}_{v'i} - \sum_{i \in \mathcal{M}} \tilde{x}_{vi} \leq T_{max} \cdot \sigma_{vv'} - \varepsilon \cdot (1 - \sigma_{vv'}).$
  $\sum_{i \in \mathcal{M}} \tilde{x}_{vi} + p_v - \sum_{i \in \mathcal{M}} \tilde{x}_{v'i} \leq T_{max}(2 - \sigma_{vv'} - \sum_{i \in M} \psi_{vv'i}).$

D. Communication resource constraint reformulation

- $\tilde{y}_{e'b} - \tilde{y}_{eb} \leq T_{max} \cdot \sigma_{ee'} - \varepsilon \cdot (1 - \sigma_{ee'}).$
  $\tilde{y}_{eb} + q_e - \tilde{y}_{e'b} \leq T_{max} \cdot (2 - \phi_{ee'} - \chi_{ee'b}).$

Proposed Optimal Scheduling Scheme

Technique-3: Constraints Linearization.

E. Precedence constraint reformulation

- Task precedence constraints: $\sum_{i\in\mathcal{M}}\tilde{x}_{vi}+p_v\leq\sum_{k\in\mathcal{K}\cup\{b,c\}}\tilde{y}_{(uv),k}$.
- Network flow precedence constraints:

$$\sum_{k\in\mathcal{K}\cup\{b,c\}}\tilde{y}_{(uv),k}+q_{uv}y_{(uv),b}+\check{q}_{uv}\sum_{k\in\mathcal{K}}y_{(uv),k}$$

$$+r_{uv}y_{(uv),c}+\sum_{i\in\mathcal{M}}\tilde{x}_{vi}\leq\sum_{i\in\mathcal{M}}\tilde{x}_{vi},$$

  where $y_{(uv),b}+\sum_{k\in\mathcal{K}}y_{(uv),k}+y_{(uv),c}=1$.
- Additionally, since if the adjacent tasks of an edge (u,v) are assigned to the same rack, the intermediate data will be transferred locally without occupying network resources. **The coupling constraints between task assignments and data transfers** is written as:

$$\sum_{i\in M}\psi_{uvi}=y_{(uv),c},\forall(u,v)\in\mathcal{E}.$$

Proposed Optimal Scheduling Scheme

Finally Reformulated Problem (MILP)

$$\begin{aligned} \textbf{RP} : \min\ & C_{max} \\ \text{s.t. } & (11) - (26), \\ & T_{max} \geq C_{max} \geq T_{min} \geq \sum_{i \in \mathcal{M}} \check{x}_{vi} + p_v, \forall v \in \mathcal{V}. \end{aligned}$$

As a result, we transform the MINLP into a MILP with the help of its bounds and the generalized data transfer model, thus the **OP** can be solved by solving **RP**. Note that, **OP** and **RP** are equivalent since the satisfaction of all constraints in **RP** indicate the satisfaction of the ones of **OP**, and vice versa. **RP** can be optimally solved by the Branch and Bound (**B&B**) algorithm, making it possible to jointly schedule jobs and wireless transceivers efficiently.

Proposed Optimal Scheduling Scheme

Technique-4: Decomposition and Acceleration

$$
\begin{aligned}
\textbf{RP}: \min\ & C_{max} \\
\text{s.t. }\ & (11)-(26), \\
& T_{max} \geq C_{max} \geq T_{min} \geq \sum_{i \in \mathcal{M}} \check{x}_{vi} + p_v, \forall v \in \mathcal{V}.
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \textbf{FP}: \text{find }\ & \mathbf{x}, \tilde{\mathbf{x}}, \mathbf{y}, \tilde{\mathbf{y}} \\
\text{s.t. }\ & \text{constraints in } \textbf{RP}, \\
& \text{set } T_{max} = \ell, \text{and } \ell \in [T_{min}, T_{max}],
\end{aligned}
$$

where ℓ is the updated upper bound of C_{max}.

- During each iteration, we solve the **FP** at midpoint $\ell = \frac{T_{min}+T_{max}}{2}$ and bisect the interval. Repeat this procedure $log_2(T_{max} - T_{min})$ times until the width of the interval is small enough.

Simulation Results

A. Job types

- Simple MapReduce workflows (1/3), one-stage MapReduce workflows (1/3), random workflows (1/3).

B. Parameters

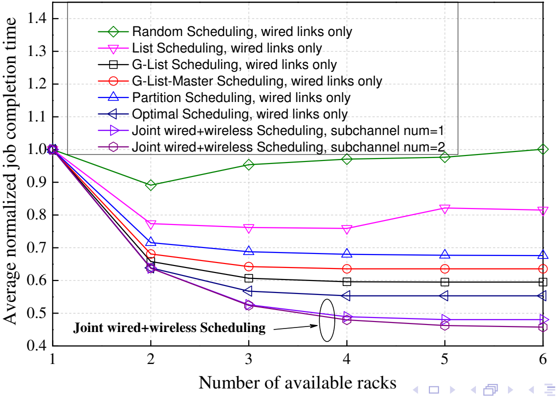
- Processing time of computing tasks : uniformly chosen from $[1, 100]$.
- Network factor ρ : The ratio between the average data transfer time and the average processing time, which is defined to set data transfer time. The larger the network factor, the higher the data size. Ranging from $[0.1, 10]$. It is around 0.5 in production scenarios.

C. Baseline algorithms

- Random Scheduling scheme
- List Scheduling scheme
- Partition Scheduling scheme
- Generalized List (G-List) Scheduling scheme
- G-List-Master Scheduling scheme

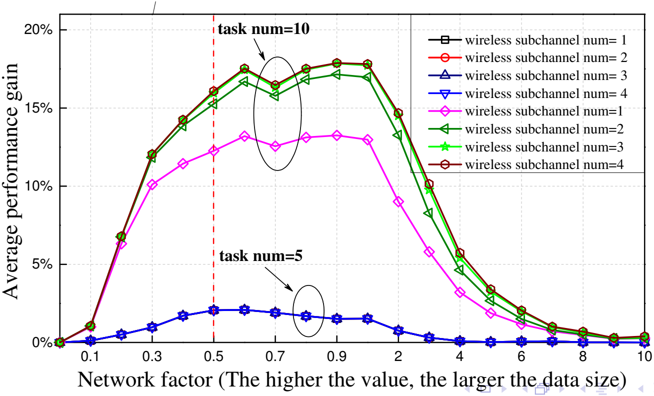
Simulation Results

1. Fix the network factor $\rho = 0.5$, half of time is spent on data transfers.
2. The task number of each job is chosen from $[5, 10]$.
3. When the racks are insufficient, the performance gain is small.
4. As the available rack number increases, adding wireless sub-channels can reduce the job completion time by up to 10%.



Simulation Results

1. Fix the available rack number $= |\mathcal{V}|$, and vary ρ from 0.1-10 to show the impact of the increased data sizes on the performance gain.
2. With the increase of ρ, the gain increases at first and then decreases.
3. With a fixed ρ (e.g.,red dash line), the larger the task num, the higher the performance gain achieved by wireless bandwidth augmentation.



Conclusion

1. By modeling the joint job scheduling and bandwidth augmentation problem in hybrid DCN, we observe the problem is a **complex MINLP**, which is **not solvable** by existing optimization methods.
2. An optimal scheduling scheme is achieved by **transforming** the MINLP into an equivalent problem, with the help of bounds estimation, the revised data transfer representation, non-linear constraints decoupling and linearization, and the Branch and Bound.
3. Simulation results show the performance gain introduced by wireless augmentation depends on multiple factors, especially the data size. Under the setting of production scenario, it can averagely reduce the job completion time by up to 10% compared with existing solutions.
4. In our future work, we will study job scheduling problems that involving more real world constraints for **on-line scenarios**.

Thank you!

bqguo@stu.xidian.edu.cn

Preprint version of this work: https://arxiv.org/abs/2209.11485

Feel free to contact us if you may have any questions.