Towards Location-aware Joint Job and Data Assignment in Cloud Data Centers with NVM

Xin Li[1], Jie Wu[2], Zhuzhong Qian[3], Shaojie Tang[4], and Sanglu Lu[3]

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics

[2]Center for Networked Computing, Temple University

[3]State Key Laboratory for Novel Software Technology, Nanjing University

[4]Naveen Jindal School of Management, The University of Texas at Dallas

Outline

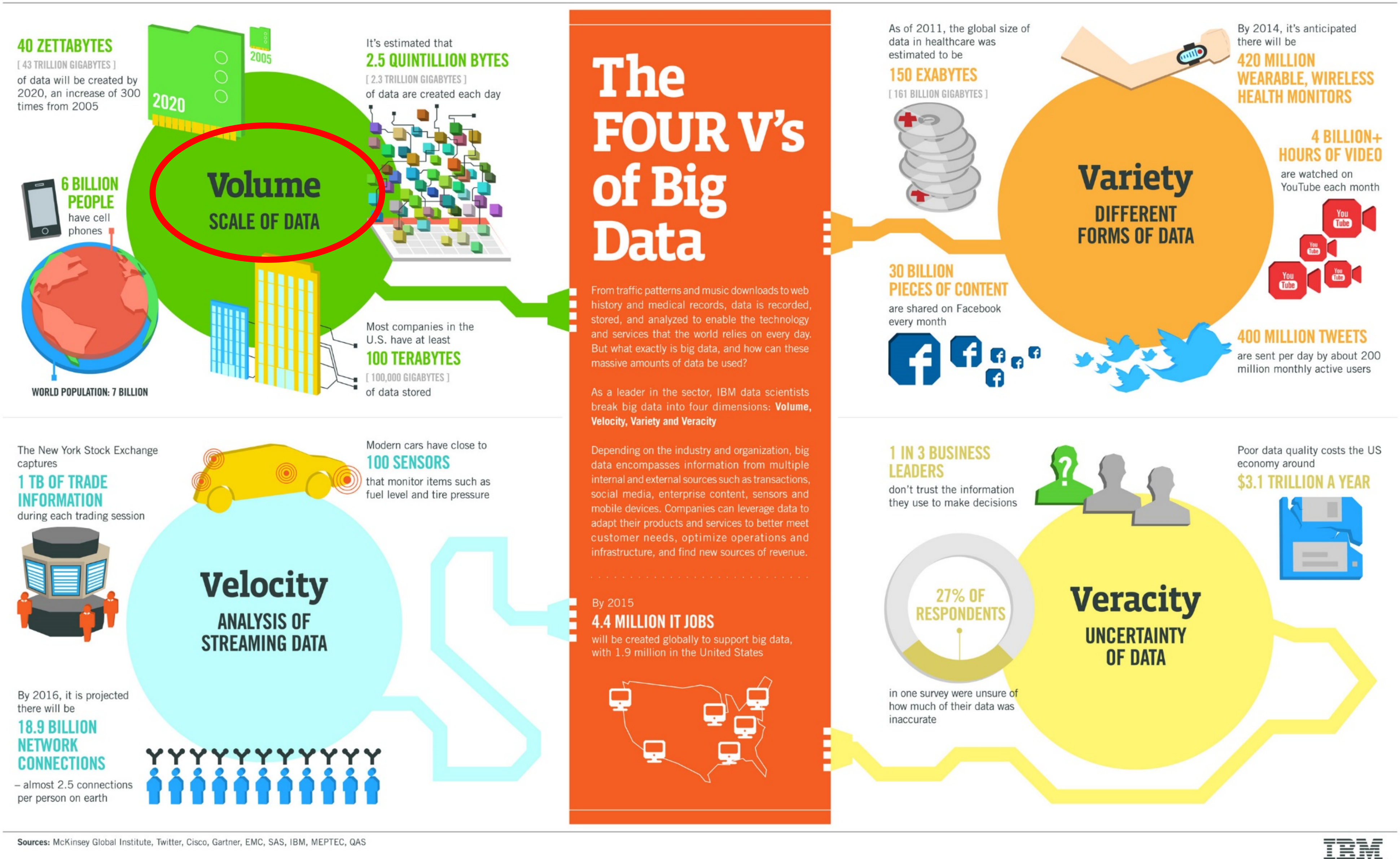
- Motivation
- Problem Statement
- Main Idea
- Performance Evaluation
- Conclusion

Motivation

- We are in the big data era.
- Timely data analysis is important to support better predictions and decision-making.
- How to reduce the data-processing time?

Loading data from disk to memory, CPU.

Four V's of Big Data



Non-Volatile Memory

- DRAM is approaching scalability limits

- NVM (Non-Volatile Memory) can achieve storage-class memory capacity, which is expected to be equipped in future data center.

- This provides faster data access speed, and it motivates us to reconsider the joint job and data assignment problem in data centers with NVM.

Data Locality

- For data-intensive jobs, the job execution time is mainly determined by the data processing time.
- Data locality
  - The job and its input data are located on the same server.
  - It could be better to preload the data in NVM for batched jobs.

How to assign the job and data jointly to minimize the makespan?

Problem Statement

- Scenario
  - For a data center that consists of uniform servers, jobs share both a data set and resources.
  - Each server hosts one job per time slot
    - It is easy to extend our result to a case with multiple jobs.
  - Each job has the same execution time with data locality.
    - The map tasks or reduce tasks of a job in MapReduce have similar execution times.

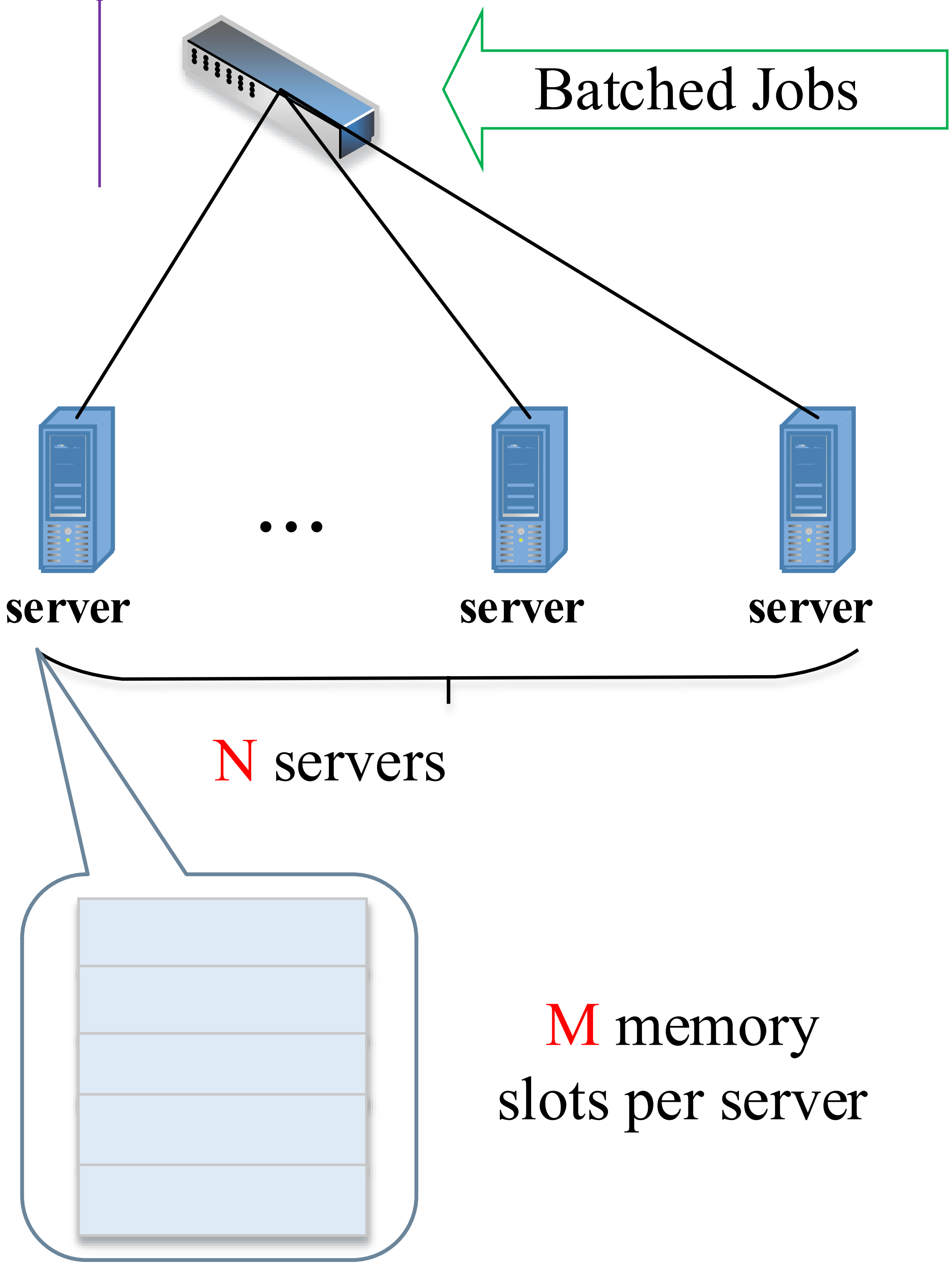
Problem Statement

- Notations
  - N: the number of uniform servers
  - M: the number of memory slots in each server
  - K: the number of data blocks
  - $< J_0, D_0 >$: D_0 is the input data for job J_0
    - Given $< J_i, D_j >$, let $f_i = D_j$

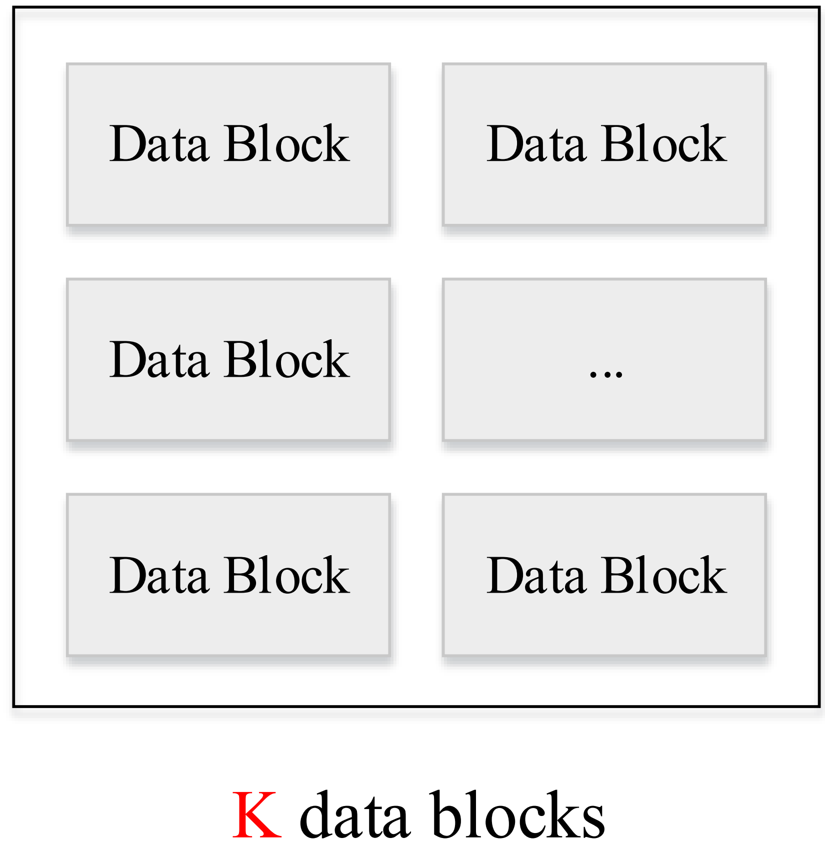
$$\pi(\mathcal{J}_i, \mathcal{S}_j) = \begin{cases} 1, & \text{Job } \mathcal{J}_i \text{ is assigned to server } \mathcal{S}_j; \\ 0, & \text{otherwise.} \end{cases}$$

$$\pi(\mathcal{D}_i, \mathcal{S}_j) = \begin{cases} 0, & \text{there is no replica of } \mathcal{D}_i \text{ on } \mathcal{S}_j; \\ 1, & \text{otherwise.} \end{cases}$$

Scenario



Batched Jobs
server
server
server
N servers
M memory
slots per server



Data Block
Data Block
Data Block
...
Data Block
Data Block
K data blocks

Problem Statement

- Given a data center consisting N uniform servers with a memory capacity of M slots and a set of jobs.
- The problem can be formulated as:

$$\textit{min.} \quad \max_{1 \le j \le \mathcal{N}} \left\{ \sum_{i=1}^{\mathcal{L}} \pi(\mathcal{J}_i, \mathcal{S}_j) \right\}$$

$$\textit{s.t.} \quad (1) \sum_{i=1}^{\mathcal{K}} \pi(\mathcal{D}_i, \mathcal{S}_j) \le \mathcal{M}, 1 \le j \le \mathcal{N}$$

$$(2) \; \pi(\mathcal{J}_i, \mathcal{S}_j) \le \pi(f_i, \mathcal{S}_j), 1 \le i \le \mathcal{L}, 1 \le j \le \mathcal{N}$$

Problem Analysis

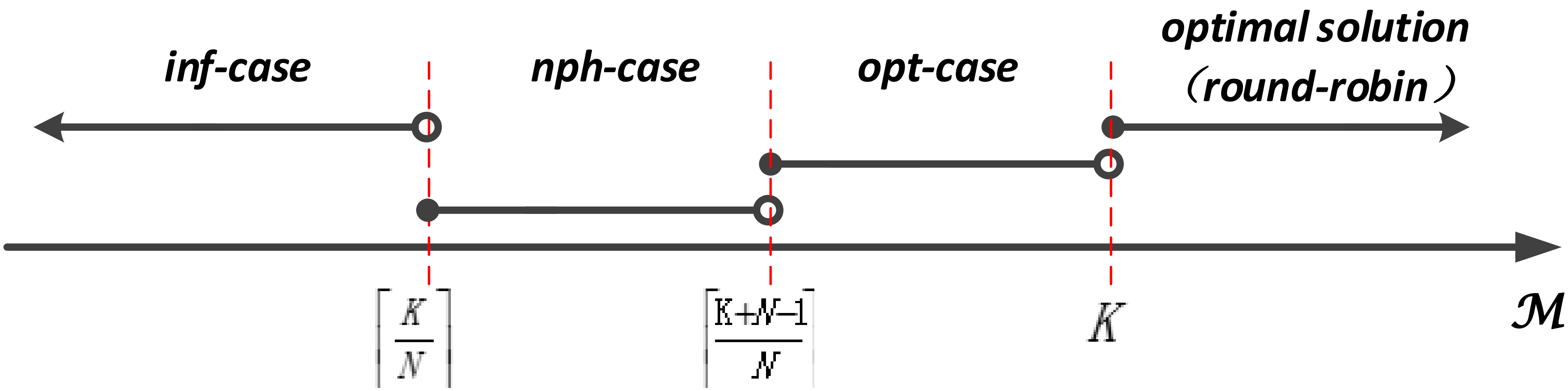
- Theorem: The joint job and data assignment problem is NP-hard.
  - Lemma: The equal-size subset-sum problem is NP-hard.
  - The problem can be reduced from the equal-size subset-sum problem in Lemma.

Problem Analysis

- Case 1: M is large enough ($M \geq K$), data locality is trivially preserved by creating one replica for all data blocks on each server.
  - Optimal solution is easy. (round-robin)
- Case 2: M the total number of memory slots is too limited ($M \times N < K$ or $M < \left\lceil \frac{K}{N} \right\rceil$)
  - *inf-case*, no feasible solution
- Case 3: $\left\lceil \frac{K}{N} \right\rceil < M < K$

Problem Analysis

- Case 3: $\left\lceil \frac{K}{N} \right\rceil < M < K$
  - *opt-case*: $\left\lceil \frac{K+N-1}{N} \right\rceil < M < K$
  - *nph-case*: $M = \left\lceil \frac{K}{N} \right\rceil$



Main Idea - Procedure

- Grouping
  - Group jobs with the same input data block.
- Sorting
  - Sort groups in ascending degree order.
- Selecting
  - Select groups step by step.
- Inserting
  - Insert the divided sub-group, and resort the groups.

Selection for *opt-case*

- Let opt be the minimized makespan, we have

$$opt \geq \varpi = \left\lceil \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{K}} d_i \right\rceil$$

- Principle: fully utilize memory slots and ensure that the workload for each server equals ϖ.
  - Partition is necessary
  - The basic idea of partitioning is to divide one group into two sub-groups with the same input data but with smaller degrees.

Condition-based Selection

- Three basic conditions for the sorted groups.

Condition 0:

$$\sum_{i=p}^{q} d_i \leq \varpi,\ q-p+1 \leq \mathcal{M}$$

Condition 1: $\Omega_1(n)$

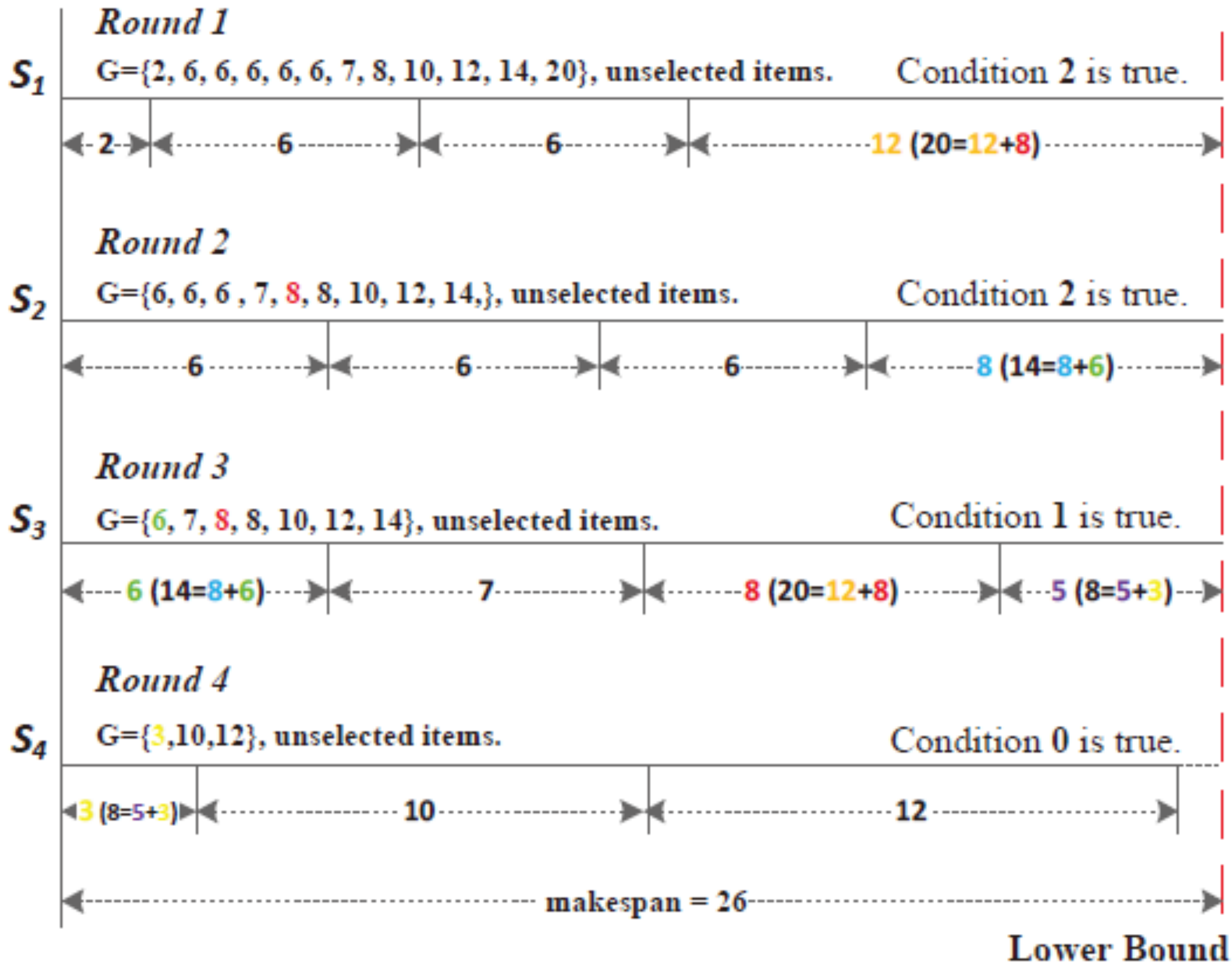
$$\sum_{i=p}^{p+n-1} d_i - \varpi = s^* \geq 0,\ \sum_{i=p}^{p+n-2} d_i - \varpi < 0, \text{ and } n \leq \mathcal{M}$$

Condition 2: $\Omega_2(m,n)$

$$\sum_{i=p}^{p+m-1} d_i + \sum_{j=q-n+1}^{q} d_j - \varpi = s^* \geq 0,$$

$$\sum_{i=p}^{p+m} d_i + \sum_{j=q-n+2}^{q} d_j - \varpi < 0, \text{ and } m+n = \mathcal{M}$$

Toy Example



Round 1
S1
G={2, 6, 6, 6, 6, 6, 7, 8, 10, 12, 14, 20}, unselected items.
Condition 2 is true.
2
6
6
12 (20=12+8)
Round 2
S2
G={6, 6, 6 , 7, 8, 8, 10, 12, 14,}, unselected items.
Condition 2 is true.
6
6
6
8 (14=8+6)
Round 3
S3
G={6, 7, 8, 8, 10, 12, 14}, unselected items.
Condition 1 is true.
6 (14=8+6)
7
8 (20=12+8)
5 (8=5+3)
Round 4
S4
G={3,10,12}, unselected items.
Condition 0 is true.
3 (8=5+3)
10
12
makespan = 26
Lower Bound

Algorithm Performance

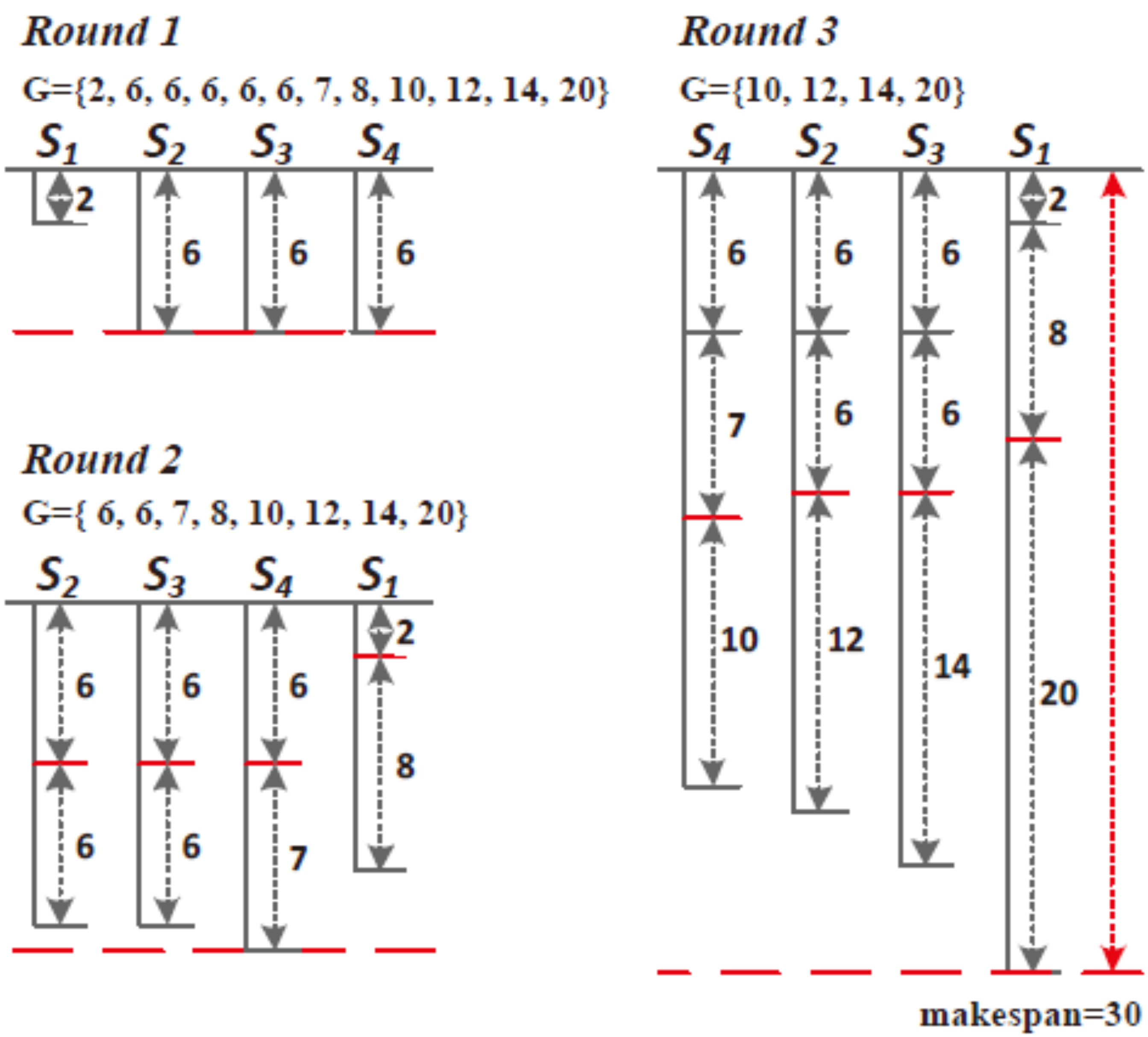
- Theorem: For the *opt-case*, i.e. $\left\lceil \frac{K+N-1}{N} \right\rceil < M < K$, the condition-based selection algorithm 1 gives the optimal assignment.

- Please find the details in our paper.

Selection for *nph-case*

- Theorem: The joint job and data assignment problem under the *nph-case* is NP-hard.
- Approximate Algorithm
  - Select one group for each server in our round.
  - There are M selection rounds.
  - One replica for each data block.
  - No group partition.

An Example



Round 1
G={2, 6, 6, 6, 6, 6, 7, 8, 10, 12, 14, 20}
S_1 S_2 S_3 S_4
2 6 6 6
Round 2
G={ 6, 6, 7, 8, 10, 12, 14, 20}
S_2 S_3 S_4 S_1
6 6 6 2
6 6 7 8
Round 3
G={10, 12, 14, 20}
S_4 S_2 S_3 S_1
6 6 6 2
7 6 6 8
10 12 14 20
makespan=30

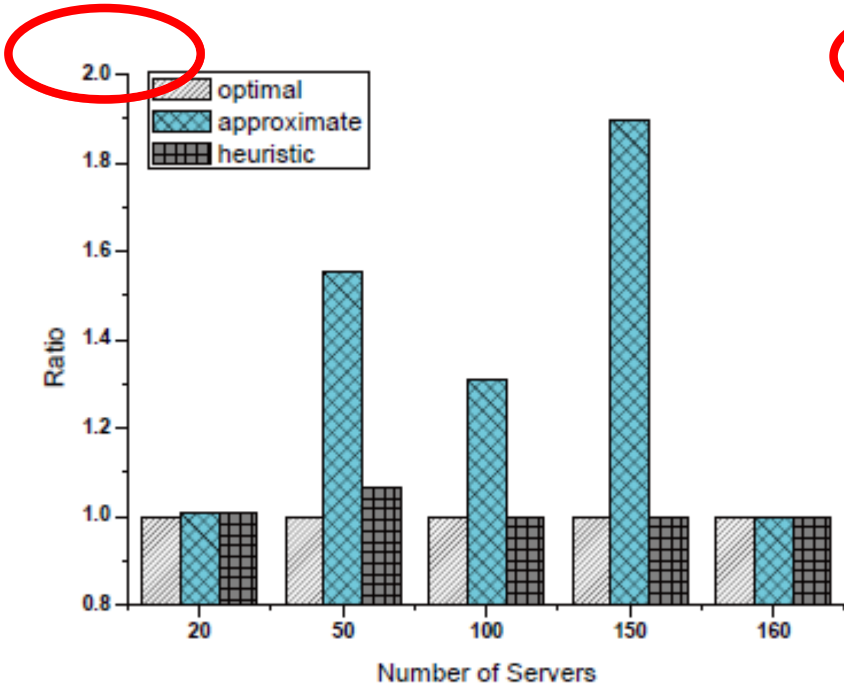
Algorithm Performance

- Theorem: For *nph-case*, the previous algorithm achieves an approximation ratio of 2.

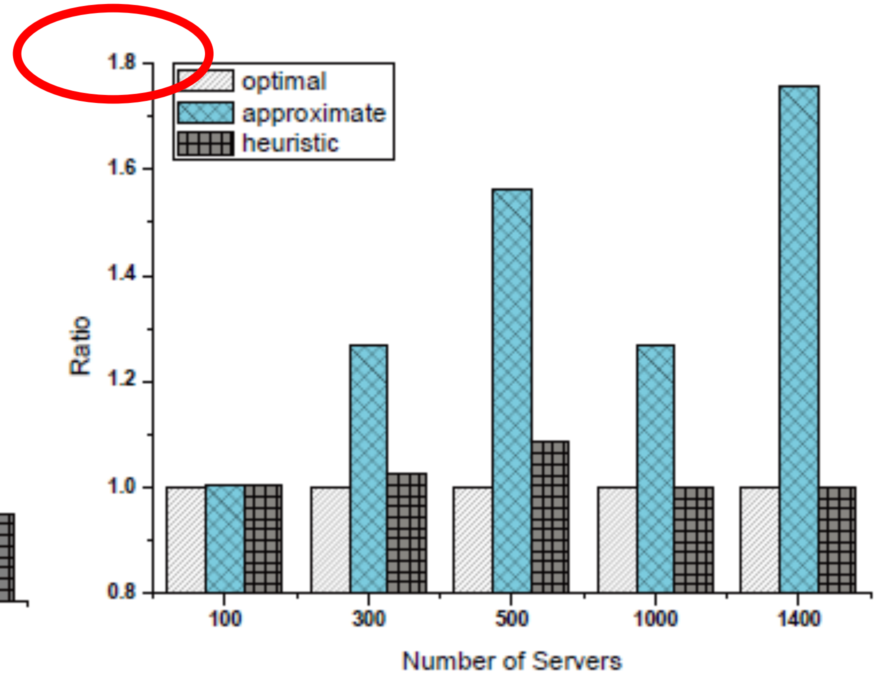
- Please find the details in our paper.

Simulation Analysis

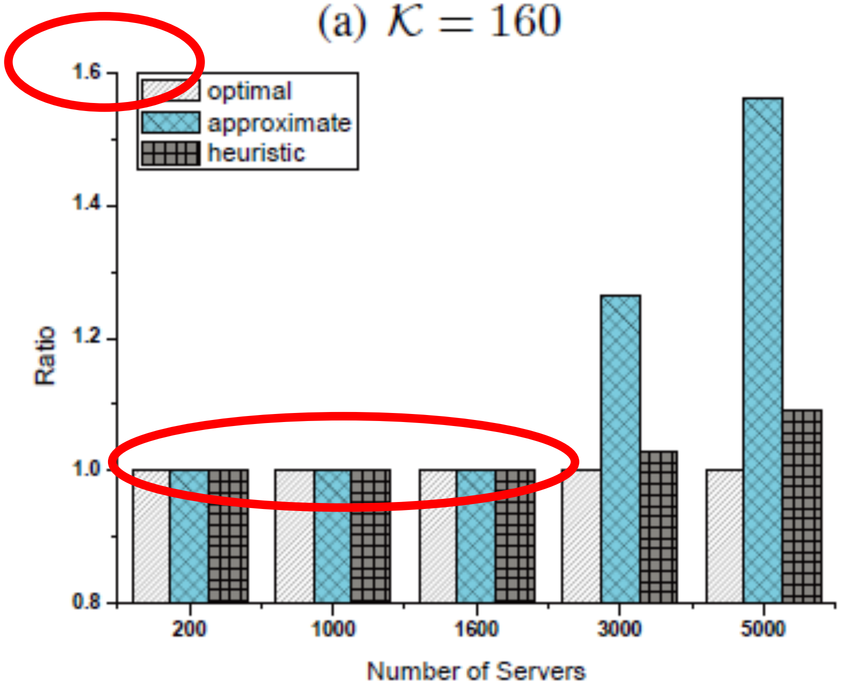
- Heuristic Algorithm
  - Assign the group with largest degree to the server with least load in greedy manner.
- Simulation Settings
  - Size of data block: 64MB
  - Size of data set: 10GB, 100GB, 1TB
    - K: 160, 1600, 16000
  - Degree of data block: random number from (0, 2000)
  - N: various values



(a) $\mathcal{K} = 160$



(b) $\mathcal{K} = 1600$



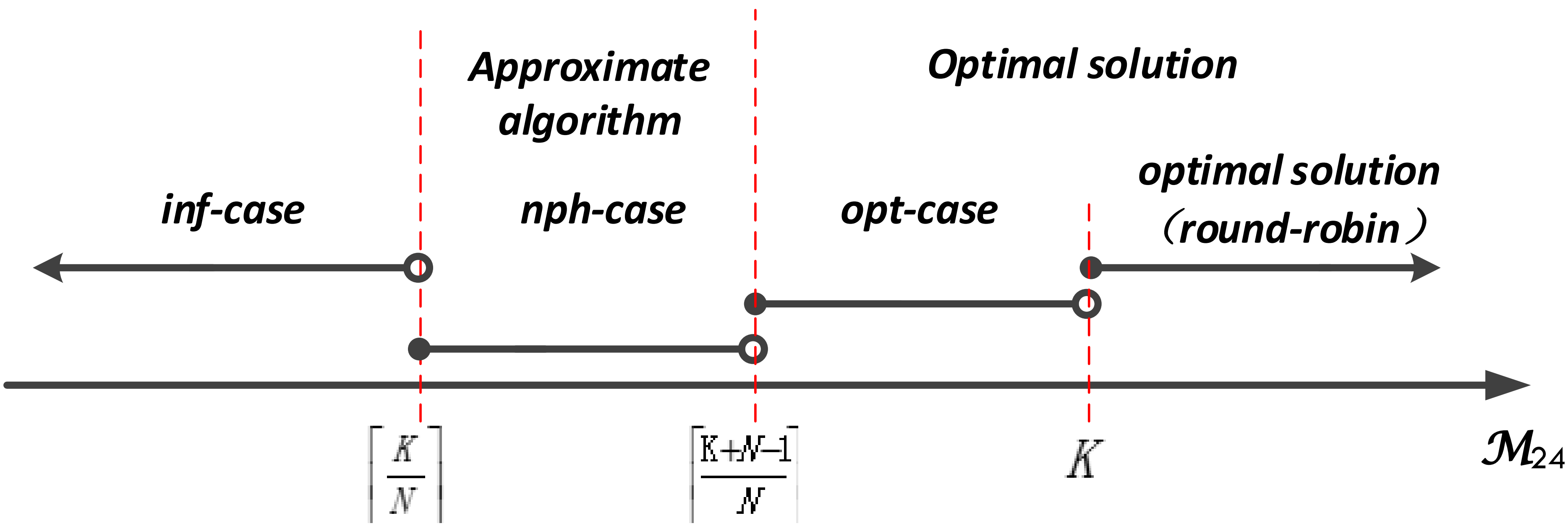
(c) $\mathcal{K} = 16000$

Simulation Results

Conclusion

- Joint job and data assignment problem for data centers with NVM.

$$0 \leq \left\lceil \frac{K+N-1}{N} \right\rceil - \left\lceil \frac{K}{N} \right\rceil \leq 1$$ Optimal solution works mostly.



Thank You!