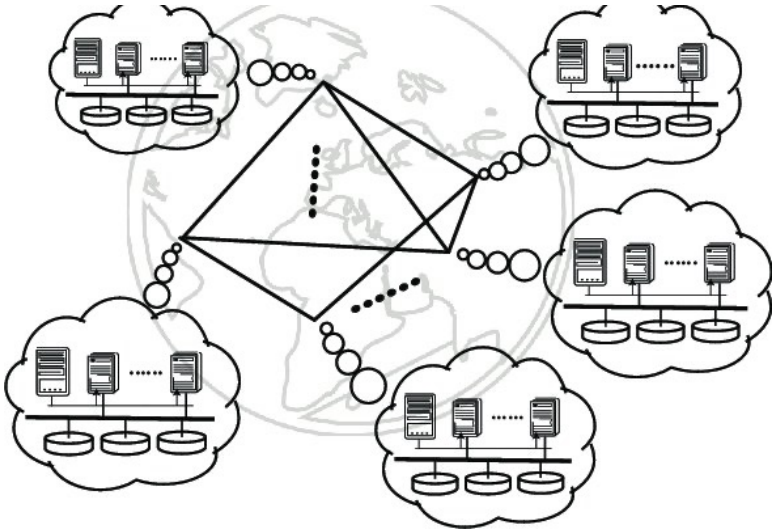
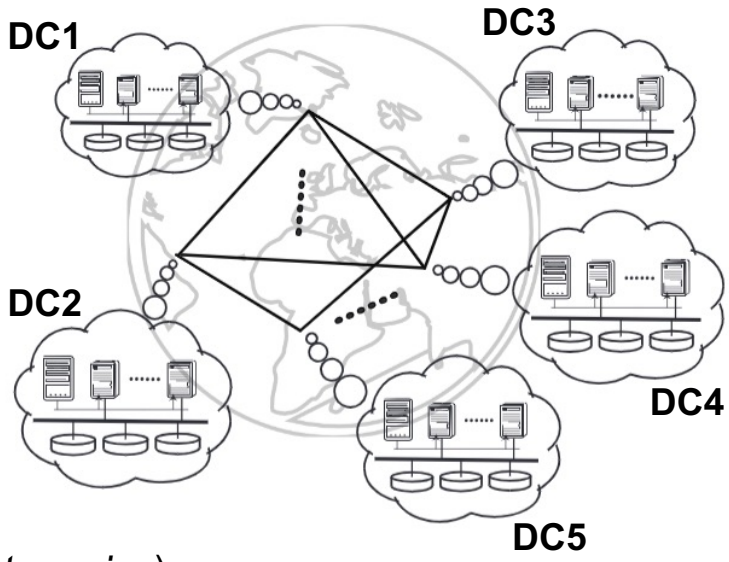
Efficient Data Management on a Cloud with Geo-Distributed Data Centers



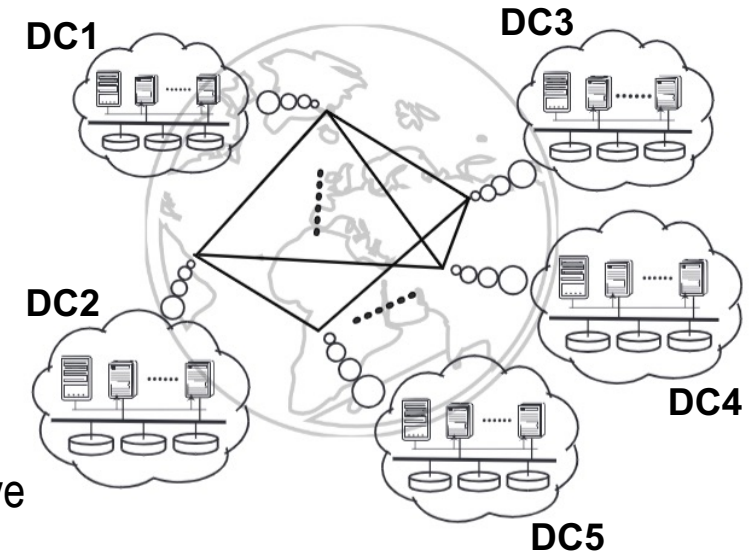
Claude TADONKI
MINES-PARISTECH / PSL

Joint work with **Laila BOUHOUCH** and **Mostapha ZBAKH**

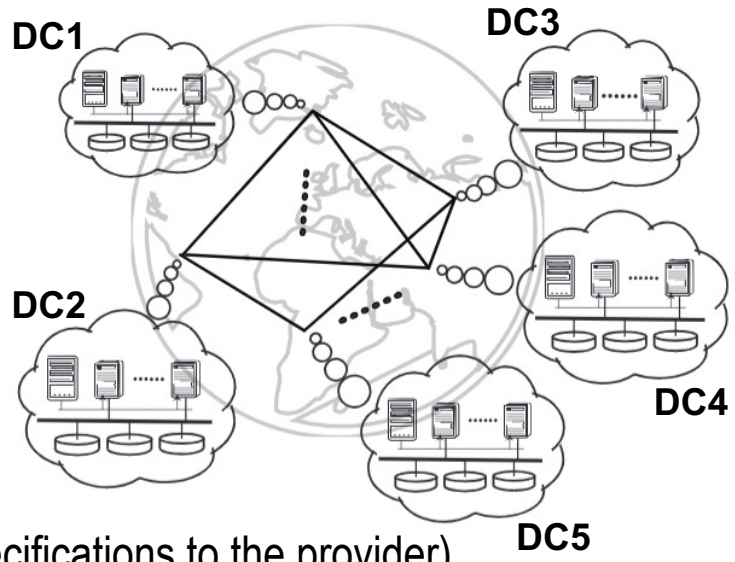
C+HPC
Cloud & High Performance Computing Laboratory
2° WCloud-HPC



- Several data centers at different geographical locations
- Typically a federation of cloud services
- The standard way to get a cloud with a larger capacity
- Good solution for applications with
  - dynamic contents and demands (e.g. *social media streaming*)
  - heavy storage and high bandwidth requirements
  - access to geo-distributed datasets (e.g. *genomics, search engine*, …)
  - very costly calculations (*federation of supercomputers*)
- The connection between the data centers are direct or indirect, thus a standard connection graph
- The model represents a single cloud system with several data centers spread over different locations



- Data are typically organized into datasets
- Datasets are exchanged between the data centers by means of explicit transfers through the network
- For large datasets and/or lot of requests, the corresponding time/cost overhead might be prohibitive
- Pure static organization might not be suitable for some contexts (*i.e. on-the-fly requests*)
- The storage capacity is likely to be reached if no specific attention is paid on data management
- How should we store and (later on) migrate the datasets across the data centers so as to minimize the overall cost of data transfers ?



- We do not consider tasks migration,
  which is a genuine alternative for optimization reasons
- We consider a complete graph,
  thus there is no indirect routing mechanism (complex!)
- We assume to act at the physical level,
  thus we stand below the virtualization (i.e. scheduling specifications to the provider)

Input 1: Tasks allocation (*i.e. for each task we are given the data center where it will be executed*)

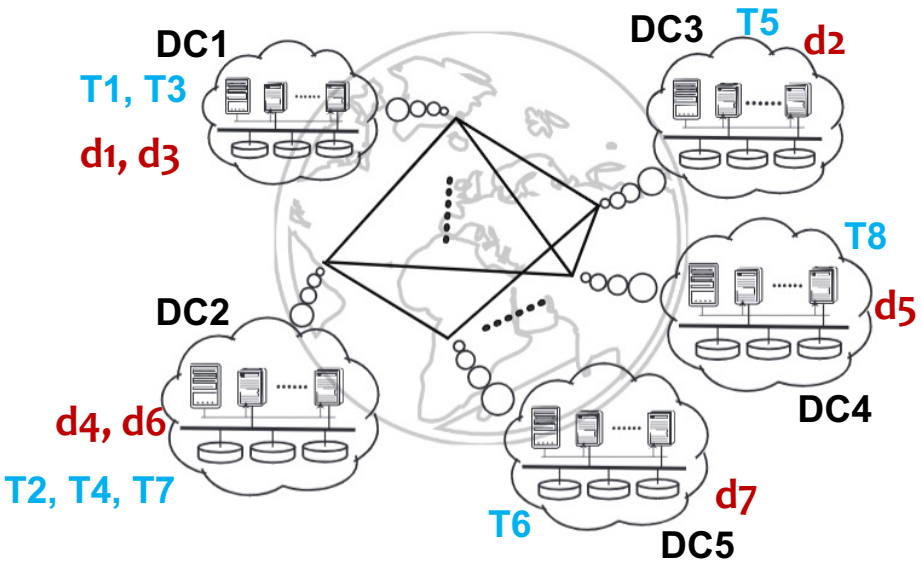
T1	T2	T3	T4	T5	T6	T7	T8
DC1	DC2	DC1	DC2	DC3	DC4	DC2	DC3

Input 2: Datasets requirement (*i.e. for each task we are given the required datasets*)

T1	T2	T3	T4	T5	T6	T7	T8
d1, d4, d6	d3	d2, d5	d6, d7	d3	d5, d7	d1, d3	d2, d4, d7

Output: Datasets allocation (*i.e. for each dataset we provide its data center location*)

d1	d2	d3	d4	d5	d6	d7
DC1	DC3	DC1	DC2	DC4	DC2	DC5



STATIC SCHEDULING

TABLE 1 Parameters and variables of our model

Inputs	
$M = (m_i)$	m_i designates the i^{th} data center
$C = (c_i)$	c_i is the storage capacity of data center m_i
$R = (r_i)$	r_i is the read speed of data center m_i
$W = (w_i)$	w_i is the write speed of data center m_i
$B = (b_{ij})$	b_{ij} is the bandwidth of the connection between data centers m_i and m_j
$\Psi = (\psi_{ij})$	ψ_{ij} is the elementary data transfer time between data centers m_i and m_j
$T = (t_i)$	t_i designates the i^{th} task
$\alpha = (\alpha_i)$	α_i is the index of the data center where task t_i is assigned to (i.e. m_{α_i})
$D = (d_i)$	d_i designates the i^{th} dataset
$V = (v_i)$	v_i is the volume of dataset d_i
$F = (f_{ij})$	$f_{ij} = 1$ if d_i is required by t_j, and 0 otherwise (F is the datasets to tasks assigments matrix)
$\tau = (\tau_{ik})$	τ_{ik} is total cost of all necessary transfers of d_i from data center m_k
$E = (e_{ij})$	$e_{ij} = 1$ if there is a copy of d_i stored in data center m_j, and 0 otherwise (E is the replication matrix)
$Q = (q_i)$	q_i is the total monetary cost of the data storage associated to task t_i (per unit of volume)
$P = (p_i)$	p_i is the total monetary cost of all data transfers associated to task t_i (per unit of volume)
Objective function	
WCC	Total workflow data transfers cost
Outputs	
$\Phi = (\phi_i)$	ϕ_i is the index of the data center where dataset d_i is (indicated to be) placed (i.e. m_{ϕ_i})
$S = (s_i)$	s_i is the index of the chosen source data center for d_i (i.e. m_{s_i})

We mainly focus on the execution time

Data centers graph: We consider a system of geographically distributed and interconnected data centers. We model the system with a graph $G = (M, w, r, B)$. $M = \{m_i, 1 \leq i \leq p\}$ is the set of data centers. Each data center m_i is characterised by its storage capacity c_i, its read speed r_i and its write speed w_i. Next, we consider matrix $B = (b_{ij})_{1 \leq i,j \leq p}$ to model the interconnection between the data centers, where $b_{ij}, 1 \leq i, j \leq p$ is the bandwidth of the connection between data centers m_i and m_j. We define matrix $\Psi = (\psi_{ij})$ as follows:

$$\psi_{ij} = \begin{cases} 0 & \text{if } i = j \\ \infty & \text{if } m_i \text{ and } m_j \text{ aren't linked} \\ \frac{1}{b_{ij}} & \text{otherwise} \end{cases} \quad (1)$$

Thus, the cost of transfering a dataset of volume v from m_i to m_j is given by

$$t(v, i, j) = v \times \psi_{ij}.$$

Definition of the tasks: We assume a fixed number of tasks to be sumitted, each of which provided with its workload (not considered here) and the required datasets for its execution. As indicated in Table 1 , we denote the group of task as $T = \{t_1, t_2, \dots, t_n\}$, where n is the number of tasks. An allocation of the tasks to the data center is provided (as input) through vector $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$, i.e. t_i is executed on data center m_{α_i}. The tasks can be executed in parallel, but each one is processed in only one data center (no task migration and no preemption).

Datasets to tasks assignment: The set of datasets is $D = \{d_1, d_2, \dots, d_m\}$, and dataset d_i has volume v_i. A task might require one or several datasets and a single dataset may be consumed by several different tasks. This led us to a $m \times n$ matrix F to model the assignment of the datasets and the tasks:

$$f_{ij} = \begin{cases} 1 & \text{if } d_i \text{ is required by } t_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Constraints: The only constraint that is considered in this paper is related to the data center storage capacity. The total volume of the datasets stored in a given data center should be lower than its capacity.

Data placement solution: This the (initial) datasets to data centers assignment that is provided as the result of a given strategy. We use vector $\Phi \in N^{|D|}$ where ϕ_i is the index of the data center that should house d_i, $i \in 1, \dots, m$. Thus

$$d_i \text{ is stored in data center } m_{\phi_i} \tag{4}$$

Data replication: We consider that a given dataset might be replicated in order to create several source alternatives. We consider a $m \times p$ matrix E to indicate the data centers that hold a copy (replica) of a dataset:

$$e_{ij} = \begin{cases} 1 & \text{if } d_i \text{ has a replica in } m_j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$WCC(W,C) = \sum_{j=1}^{|T|} \sum_{i=1}^{|D|} f_{ij} \times \tau^{(i)}(s_i, \alpha_j) \tag{6}$$

where

$$\tau^{(i)}(s_i, \alpha_j) = \begin{cases} (\frac{1}{r_{s_i}} + \frac{1}{w_{\alpha_j}} + \frac{1}{b_{s_i \alpha_j}}) \times v_i & \text{if } s_i \neq \alpha_j \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The mathematical programming formulation of our problem could be written as follows:

$$\min_{s} \sum_{j=1}^{|T|} \sum_{i=1}^{|D|} f_{ij} \times \tau^{(i)}(s_i, \alpha_j)$$
$$\text{subject to} \tag{8}$$
$$\sum_{j=1}^{|D|} v_j \delta_{is_j} \leq c_i, i = 1, \cdots, |M|$$

where δ is the Kronecker symbol (i.e. $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.). This is what we intend to solve by a heuristic approach.

Algorithm 1 Calculate data transfer time matrix

Input:

$M = (m_1, m_2, \dots, m_p)$: Set of datacenters
$T = (t_1, t_2, \dots, t_n)$: Set of tasks
$D = (d_1, d_2, \dots, d_m)$: Set of datasets
$C = (c_1, c_2, \dots, c_p)$: Capacities of the datacenters
$V = (v_1, v_2, \dots, v_m)$: Volumes of the datasets
F : Matrix of relationship between datasets and tasks
τ : Matrix of datasets transfer time over the datacenters

Output:

$\phi = (\phi_1, \phi_2, \dots, \phi_m)$: Array of final data placement

// Computation of τ matrix
for $i \in D$ **do**
  for $j \in M$ **do**
    $\tau(i, j) \leftarrow 0$
    for $k \in T$ **do**
      if $f(i, k) == 1$ **then**
        if $j \neq \alpha_k$ **then**
          $\tau(i, j) \leftarrow \tau(i, j) + (\frac{1}{r_j} + \frac{1}{w_{\alpha_k}} + \frac{1}{b_{j\alpha_k}}) \times v_i$
        end if *// Otherwise the dataset and the task are within the same datacenter*
      end if
    end for
  end for
// Sort elements of each row $\tau(i, :)$ in ascending order
  $[\sigma(i, :)] \leftarrow sort(\tau(i, :))$ *// $\sigma_i(j) = k$, the k^{th} element of $\tau(i, :)$*
end for

Task t_k runs on data center with index α_j

Iterative computation of the total cost of all transfers of d_i from m_j

Sorting here is just speed up the access to the minimum values

Algorithm 2 Algorithm for an efficient datasets placement

Input: */* Refinement of the datasets placement following a greedy approach */*

1: c: index of the current column in the transfer cost matrix
2: s: number of already placed datasets
3: $placed$: binary array indicating the selection status

4: $c \leftarrow 1$
5: $s \leftarrow 0$
// initialization of vector placed
6: **for** ($i \leftarrow 1$ *to* m) ***do***
7: $placed[i] \leftarrow 0$
8: ***end for***
9: ***while*** ($s < m$ && $c \leq p$) ***do***
10: $max \leftarrow -1$
11: $k \leftarrow -1$
12: ***for*** $i \leftarrow 1$ *to* m ***do***
13: ***if*** ($placed[i] \neq 1$) ***then***
14: $\ell \leftarrow \sigma[i, c]$ // *id of our* c^{th} *acceptable datacenter choice for dataset* d_i
15: ***if*** (($\tau(i, \ell) > max$) && ($c_\ell - v_i > 0$)) ***then***
16: $max \leftarrow \tau(i, \ell)$
17: $k \leftarrow i$
18: ***end if***
19: ***end if***
20: ***end for***
21: ***if*** ($k \neq -1$) ***then***
22: $\ell \leftarrow \sigma[k, c]$ // *id of the acceptable datacenter for dataset* d_k
23: $placed[k] \leftarrow 1$
24: $\phi[k] \leftarrow \ell$ // d_k *will be stored in datacenter* m_ℓ
25: $c_\ell \leftarrow c_\ell - v_k$ // *update of the capacity of datacenter* m_ℓ *as it receives dataset k*
26: s++
27: ***else***
28: c++
29: ***end if***
30: ***end while***
31: ***if*** ($s = m$) ***then***
32: *Data placed successfully !*
33: ***else***
34: *Problem with data placement !*
35: ***end if***

Try to place d_i if we didn't succeed at the previous iteration

We will relocate this one because of its cost (the highest)

For the next iteration we will consider the next optimal choice

It's possible ! Use default choices

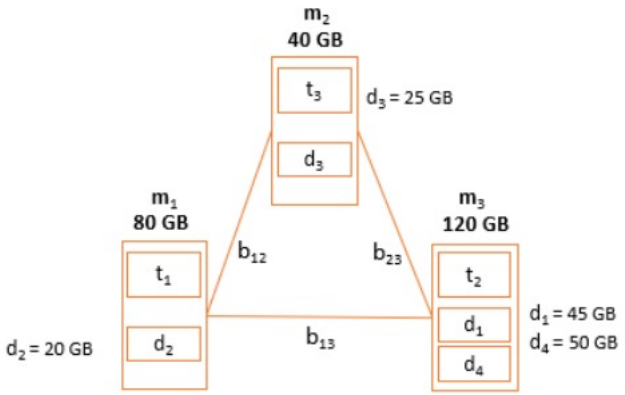


FIGURE 2 A sample data placement solution

DYNAMIC SCHEDULING

- We use datasets transfers to create replicas by deciding whether or not we should keep a received dataset as a copy(replica).

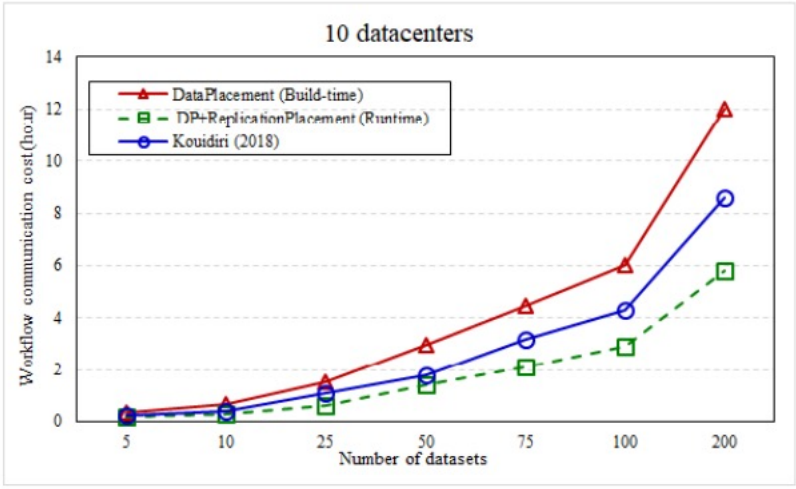
- When a data center needs a given dataset, then it gets a copy from the best location
  The array of the best locations is updated after each transfer.

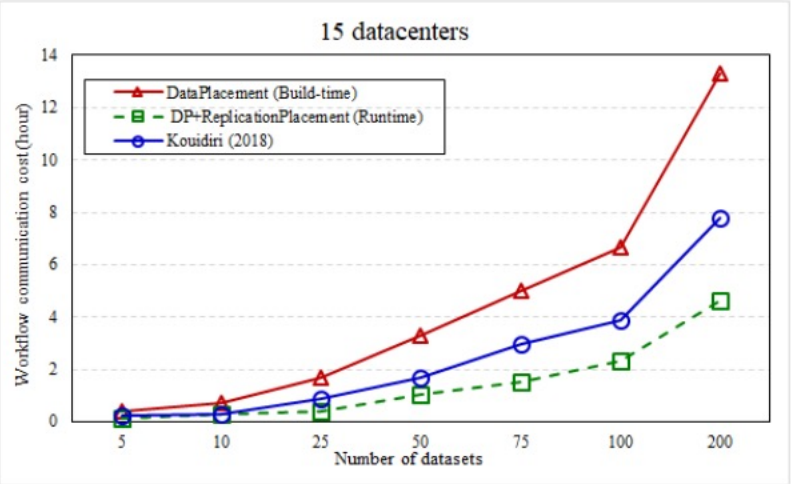
- When there not enough space to store a copy,
  We run our deletion procedure, which selects which dataset to be deleted.

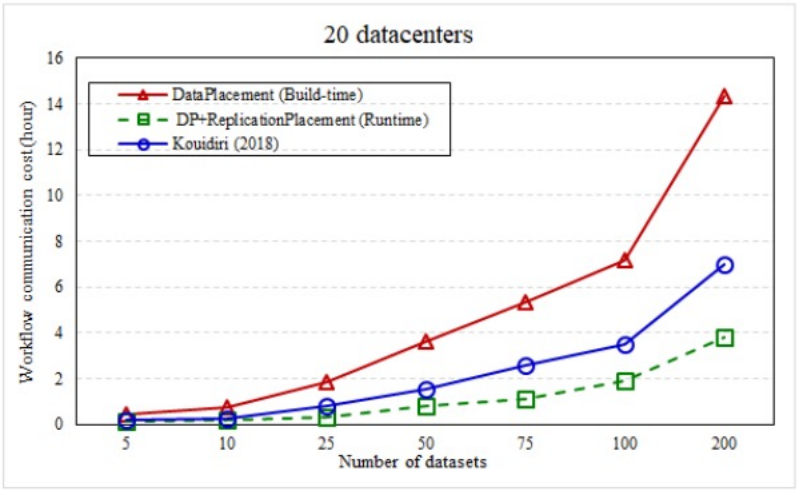
EXPERIMENTS

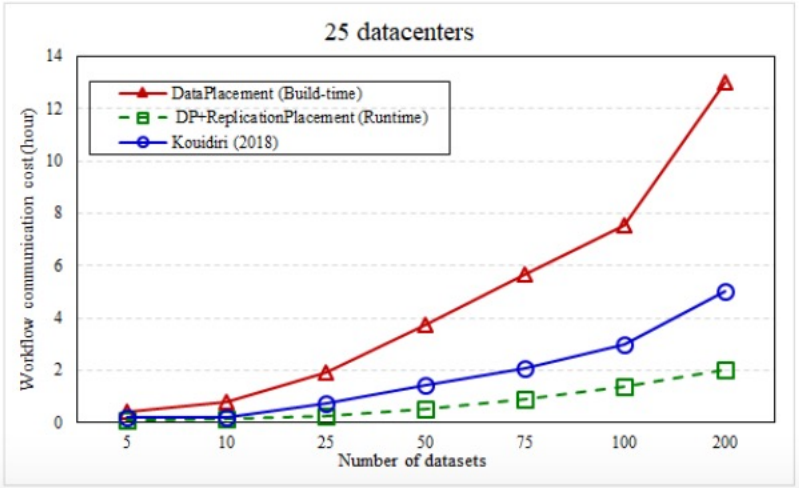
TABLE 2 Default setting for our experiment

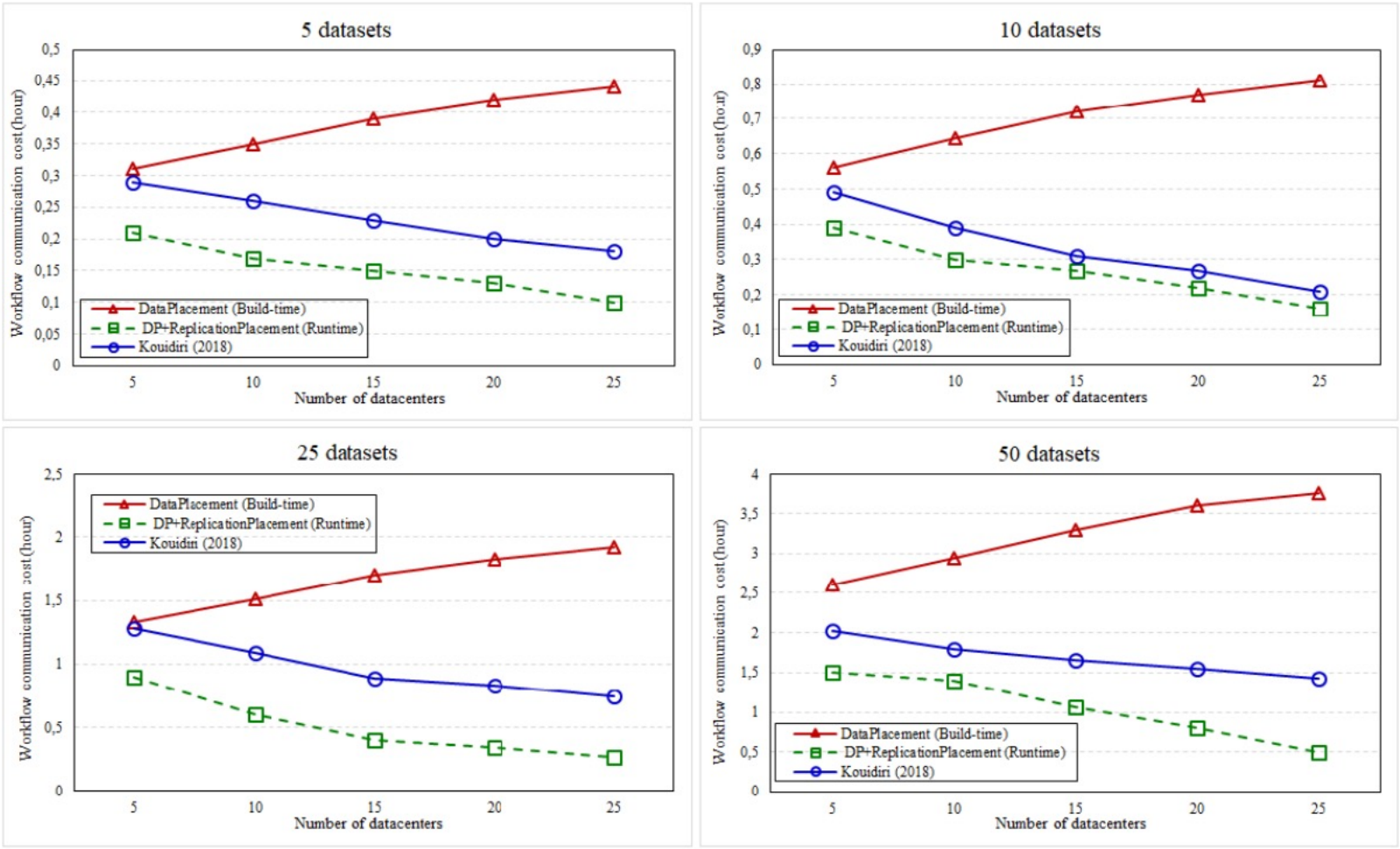
Components & Values	
# of datasets	[5,10,25,50,75,100]
Dataset size	[1TB - 100TB]
# of datacenters	[5,10,15,20,25]
Datacenter capacity	[1PB - 25PB]
Storage cost	$0.1 per GB
Transfer cost	$0.05 per GB

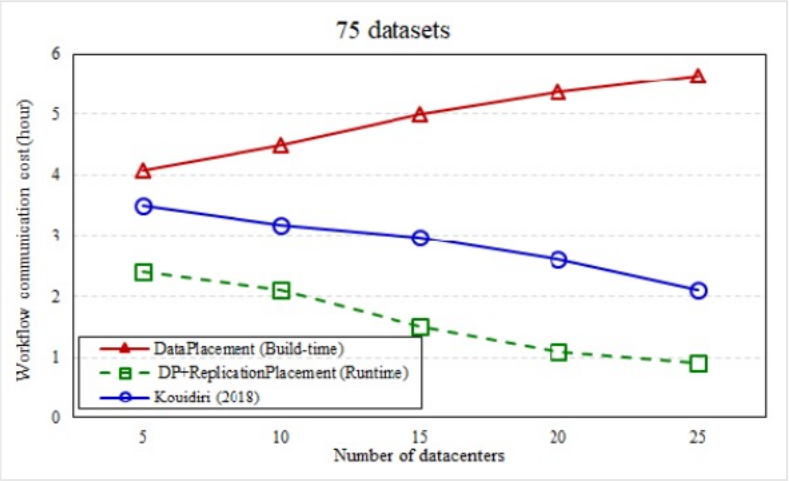


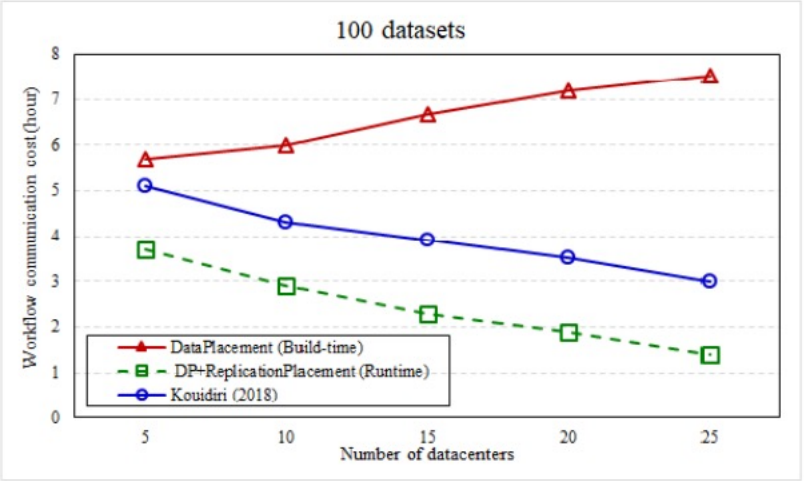

15 datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)
Workflow communication cost (hour)
Number of datasets
0
2
4
6
8
10
12
14
5
10
25
50
75
100
200

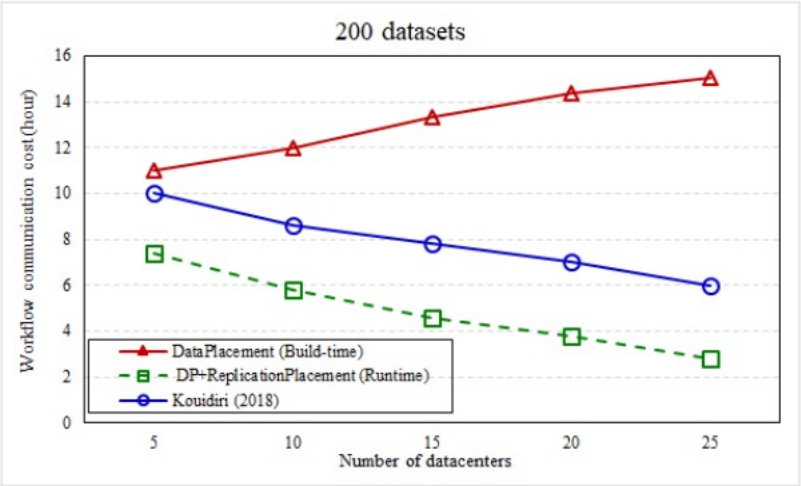

20 datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)
Workflow communication cost (hour)
Number of datasets
0
2
4
6
8
10
12
14
16
5
10
25
50
75
100
200


25 datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)
Workflow communication cost (hour)
Number of datasets
0
2
4
6
8
10
12
14
5
10
25
50
75
100
200


5 datasets
Workflow communication cost (hour)
Number of datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)
10 datasets
Workflow communication cost (hour)
Number of datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)
25 datasets
Workflow communication cost (hour)
Number of datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)
50 datasets
Workflow communication cost (hour)
Number of datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)


75 datasets
Workflow communication cost (hour)
Number of datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)


100 datasets
Workflow communication cost (hour)
Number of datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)


200 datasets
Workflow communication cost (hour)
Number of datacenters
DataPlacement (Build-time)
DP+ReplicationPlacement (Runtime)
Kouidiri (2018)

- Investigate a more fine way of using the replicas (compute the best source for each request).
- Measure the simulated data transfers time using a third-party tool..
- Consider running on a real Cloud.

END & QUESTIONS