NANJING UNIVERSITY

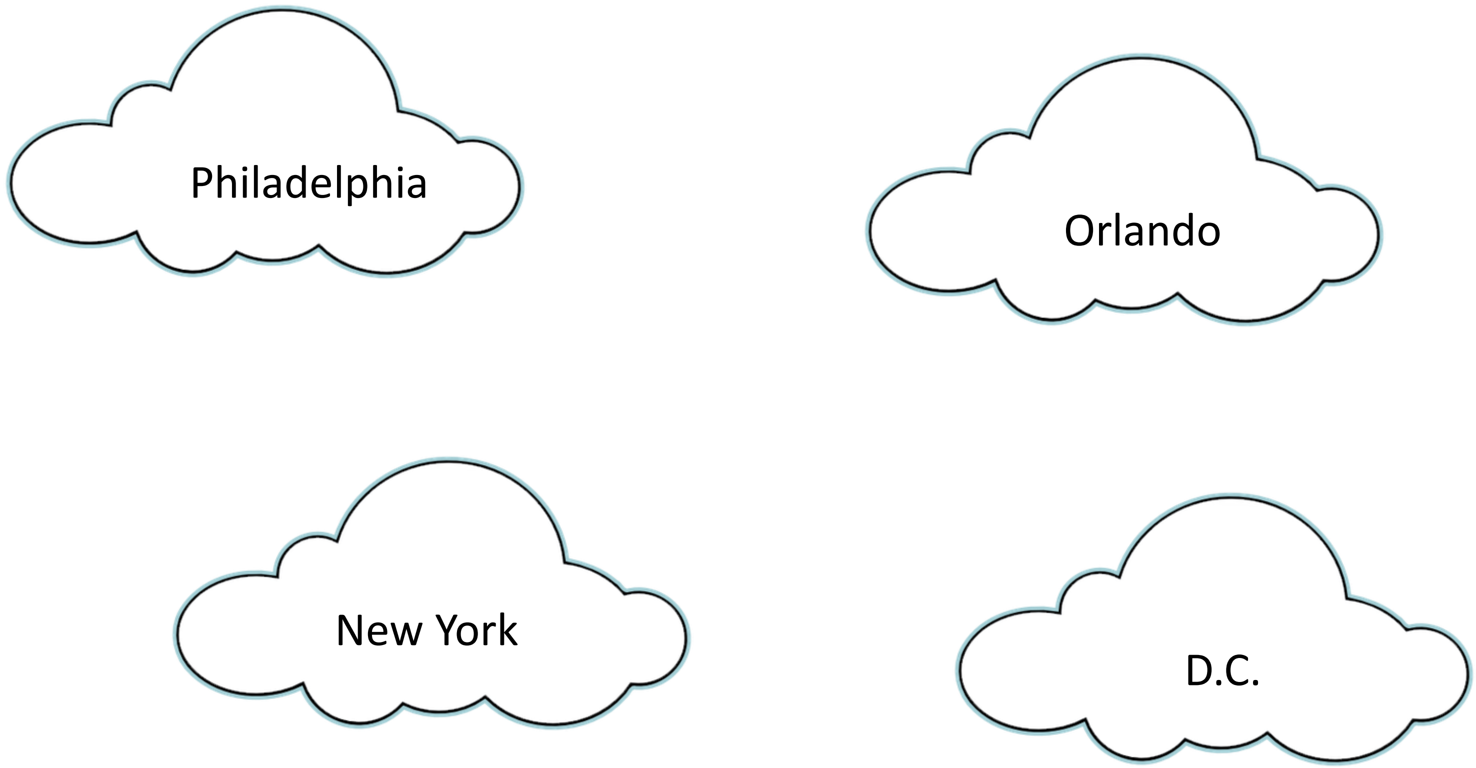
SEA: Stable Resource Allocation in Geographically Distributed Clouds

Presenter: Sheng Zhang

Sheng Zhang, Zhuzhong Qian, Jie Wu, Sanglu Lu
Nanjing University & Temple University
11-June-2014
ICC'14@Sydney

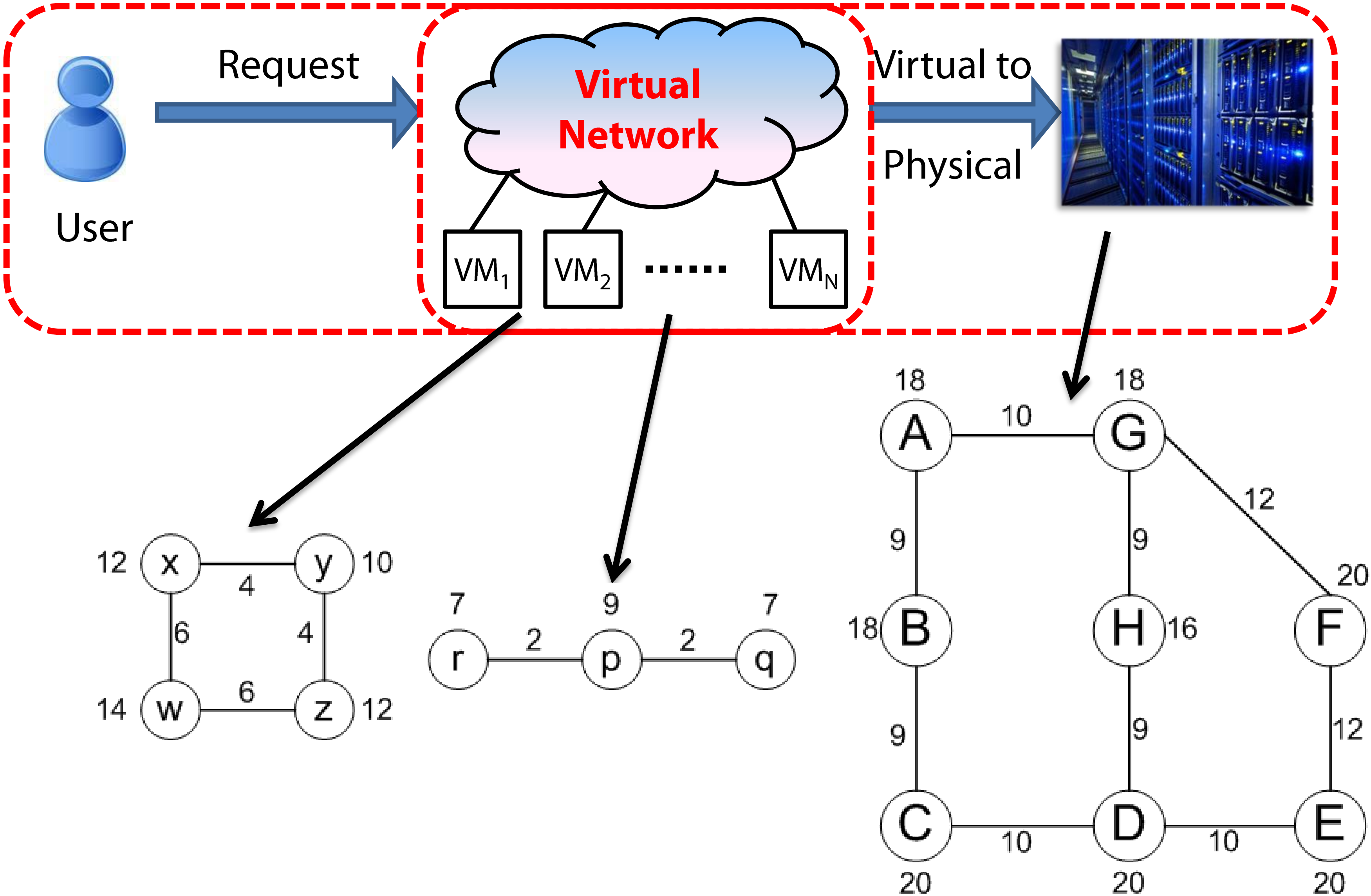
Cloud Computing

Distributed Cloud Architecture

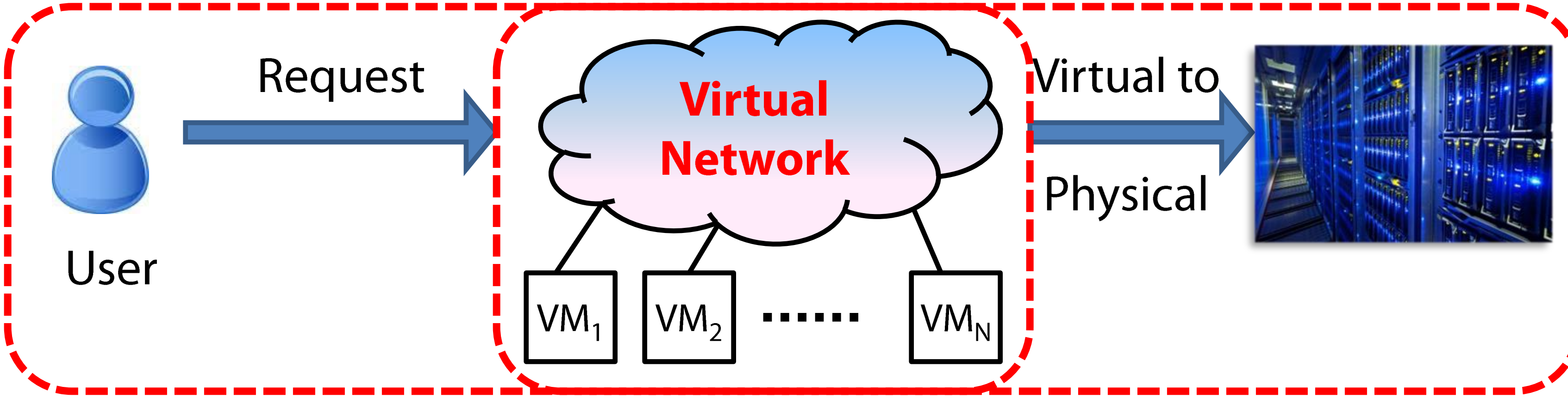


- Energy-efficient request routing by exploiting heterogeneities of carbon footprints and electricity costs
- High availability of cloud resources by locating resources closer to users

Resource Allocation in Clouds



Related Work on Resource Allocation



- Bin packing-based VM consolidation [WML11]
- Network-aware VM placememt [AL12]
- Opportunistic resource sharing [ZQWLE14]

[WML11] Consolidating virtual machines with dynamic bandwidth demand in data centers, INFOCOM 2011

[AL12] Network aware resource allocation in distributed clouds, INFOCOM 2012

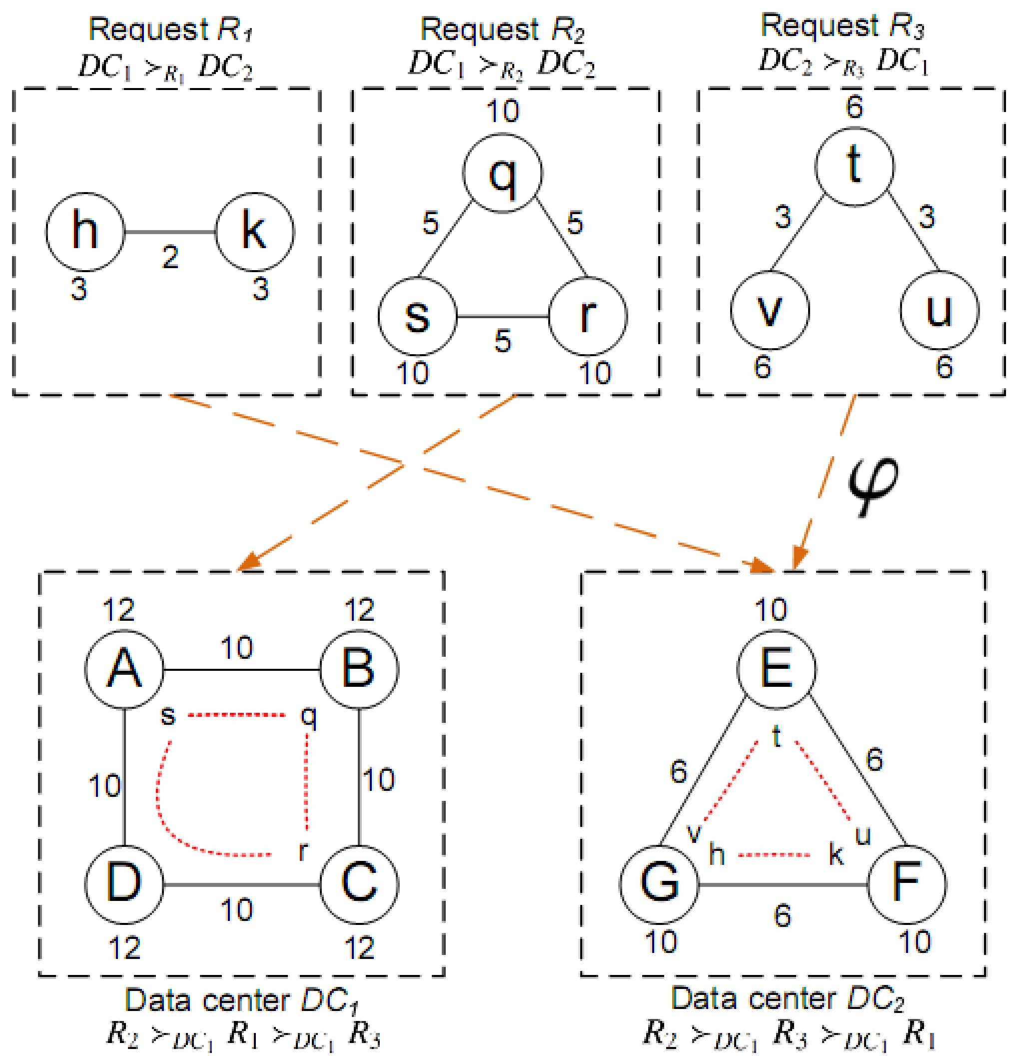
[ZQWLE14] Virtual network embedding with opportunistic resource sharing, TPDS 2014

Content

- Stable rEsource Allocation (SEA) Problem
- Solution
- Evaluation
- Conclusions

Distributed Cloud Architecture

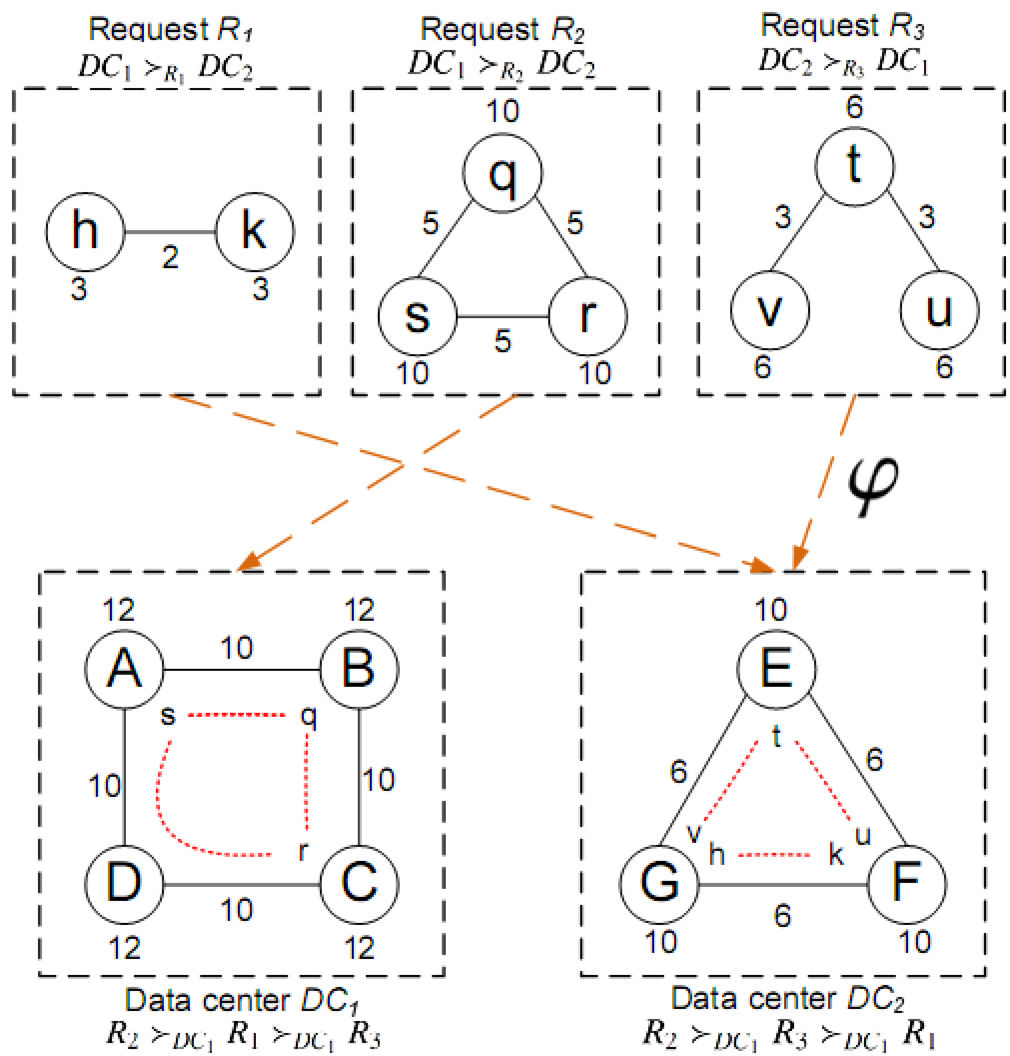
- *M* resource requests
  - $R_1, R_2, \ldots, R_M$
- $R_j = (N_j, E_j, C_j, B_j)$



- *N* small sized data centers
  - $DC_1, DC_2, \ldots, DC_N$
- $DC_i = (N_i^p, E_i^p, C_i^p, B_i^p)$

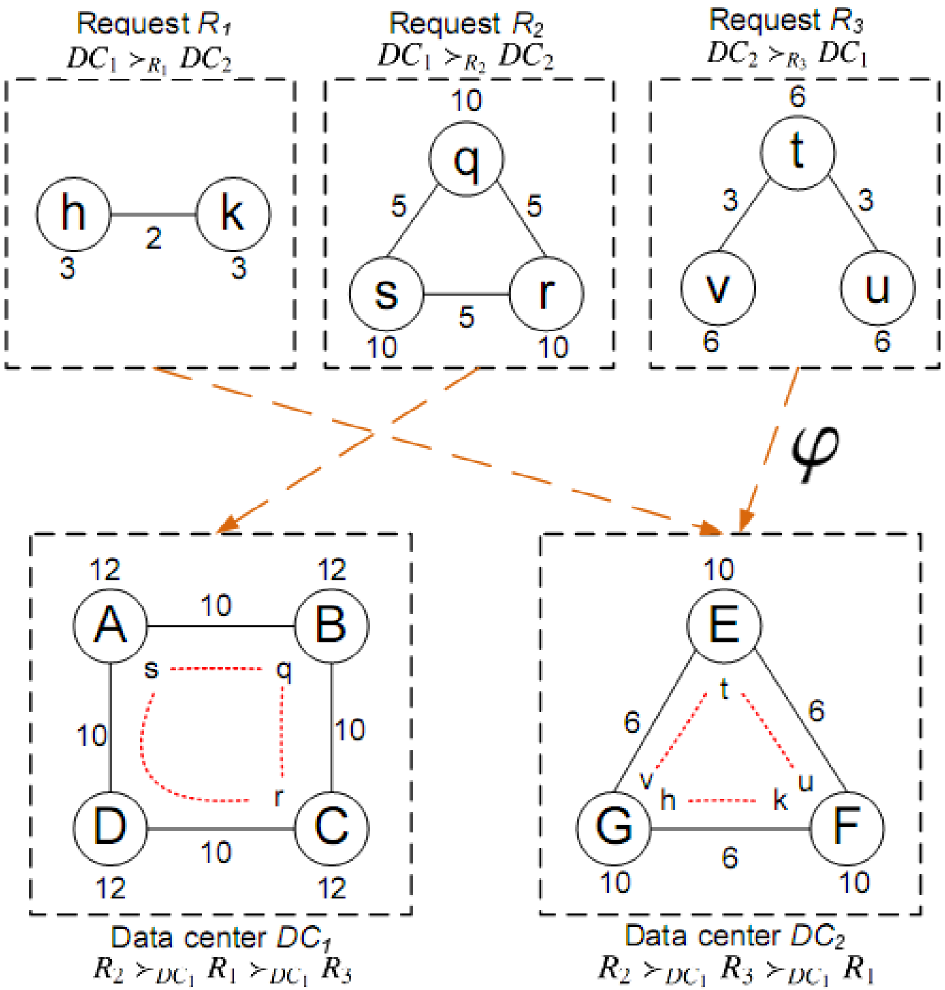
Preference List

- Each one has a preference list over the opposite entities
  - E.g., computation-intensive requests prefer high-performance DCs.



- R_j is associated with a strictly ordered list P_{R_j} containing all N data centers.
- DC_i is associated with a strictly ordered list P_{DC_i} containing all M requests

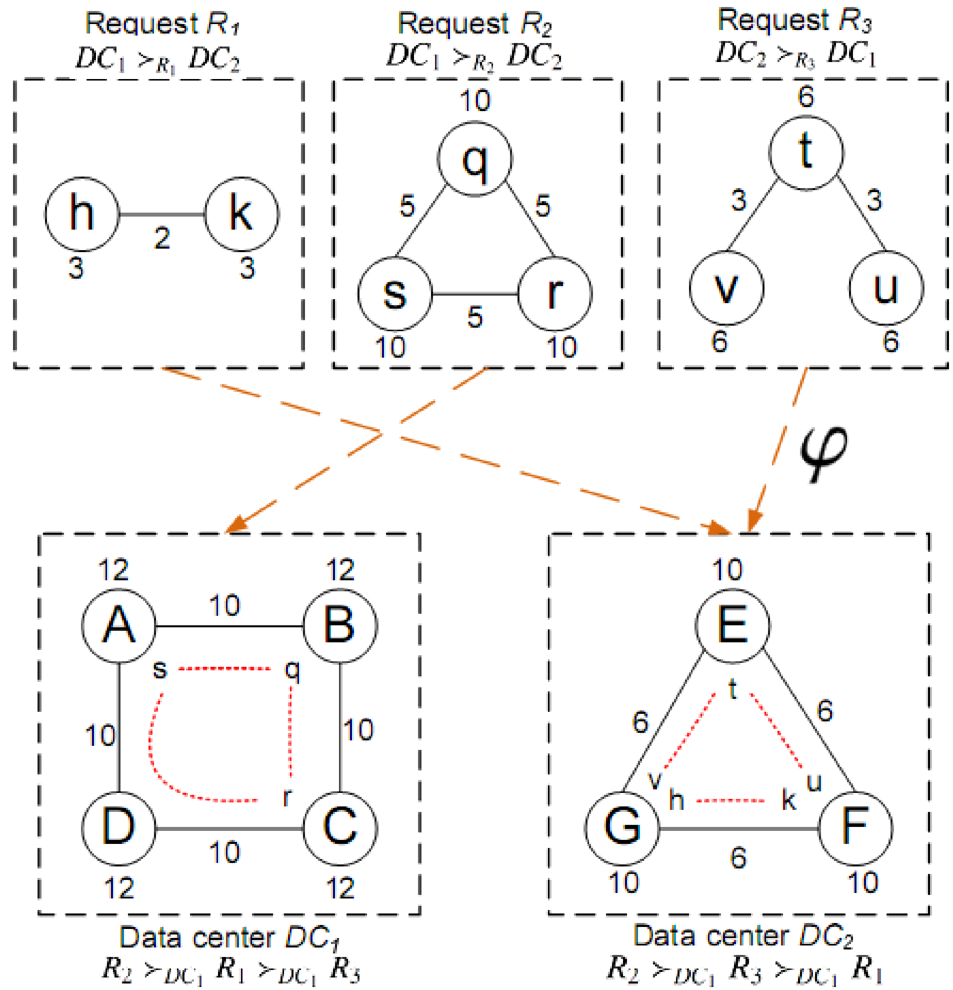
Allocation Plan



Definition 1: *(Allocation Plan).* Let DC be the set of N data centers and R be the set of M requests; an allocation plan is a many-to-one correspondence $\varphi : R \rightarrow DC$. When $\varphi(R_j) = DC_i$, we say R_j is placed in DC_i.

Fig. 1 shows an allocation plan, where $\varphi(R_1) = DC_2$, $\varphi(R_2) = DC_1$, and $\varphi(R_3) = DC_2$.

Blocking Pair

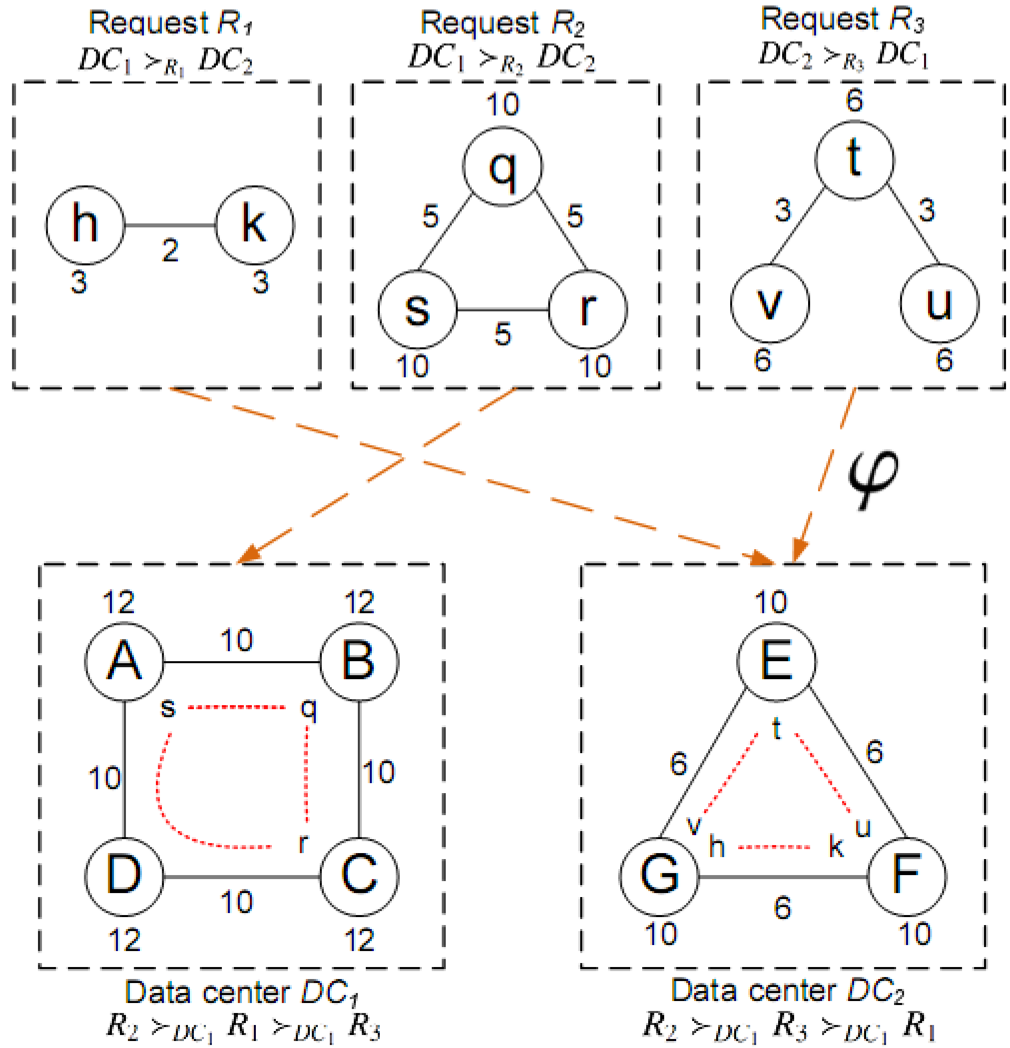


Definition 2: *(Blocking Pair)*. A pair (R_j, DC_i) of resource request and data center is a blocking pair in an allocation plan φ if *(i)* R_j is not placed in DC_i, *(ii)* R_j is either unplaced or prefers DC_i to $\varphi(R_j)$, and *(iii)* DC_i is either under-utilized or prefers R_j to the worst request placed in it.

SEA

- Given N data centers and M requests, find an allocation plan without any blocking pairs?

- The plan in the fig. is stable.
- An unstable example

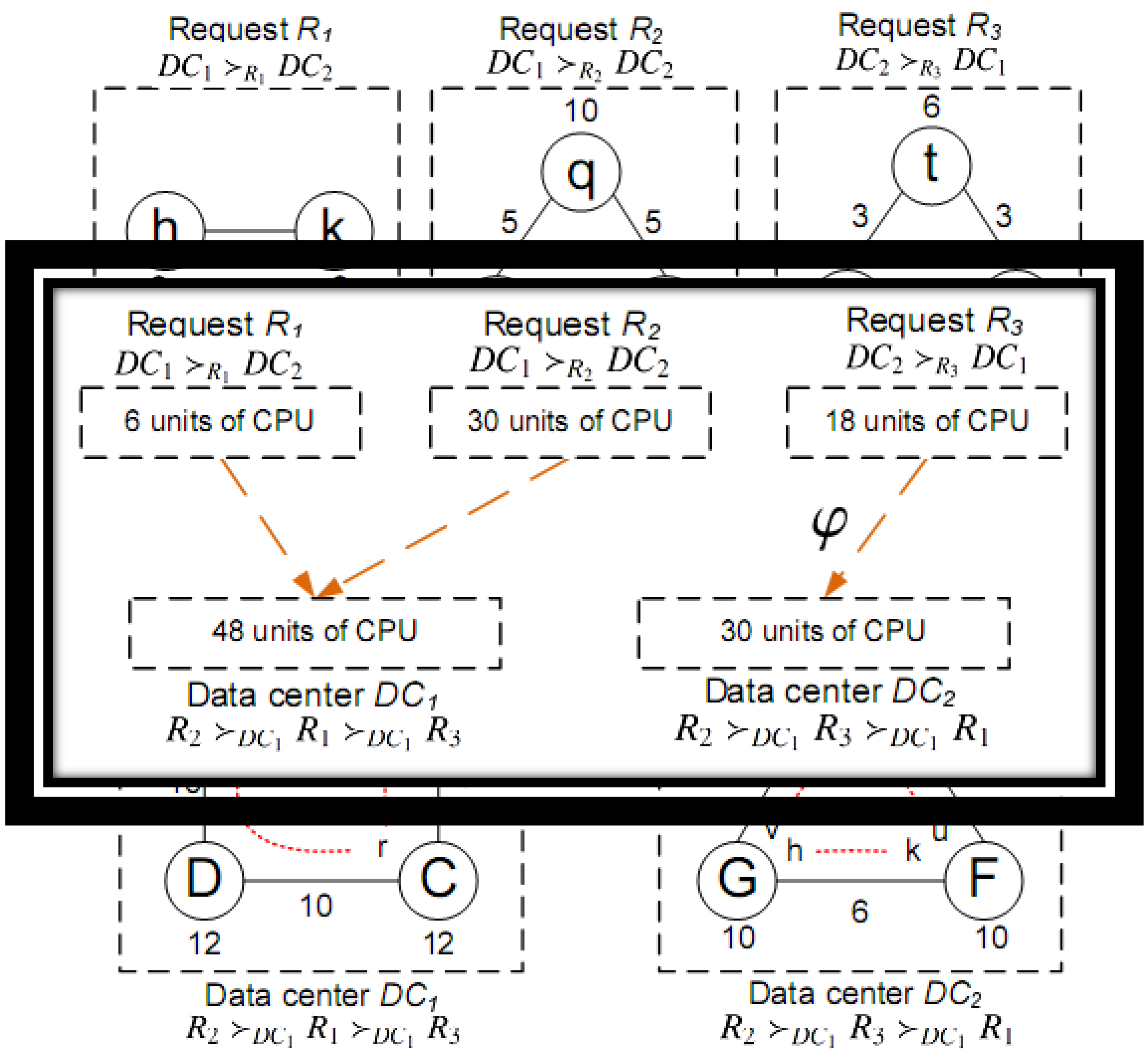


allocation plan φ': $\varphi'(R_1) = DC_1$, $\varphi'(R_2) = DC_2$, and $\varphi'(R_3) = DC_1$. In φ', (R_2, DC_1) forms a blocking pair, because (*i*) R_2 is not placed in DC_1 in φ', (*ii*) R_2 prefers DC_1 to DC_2, and (*iii*) DC_1 prefers R_2 to the worst request placed in it, *i.e.*, R_3.

Content

- Stable rEsource Allocation (SEA) Problem
- Solution
- Evaluation
- Conclusions

1-D SEA



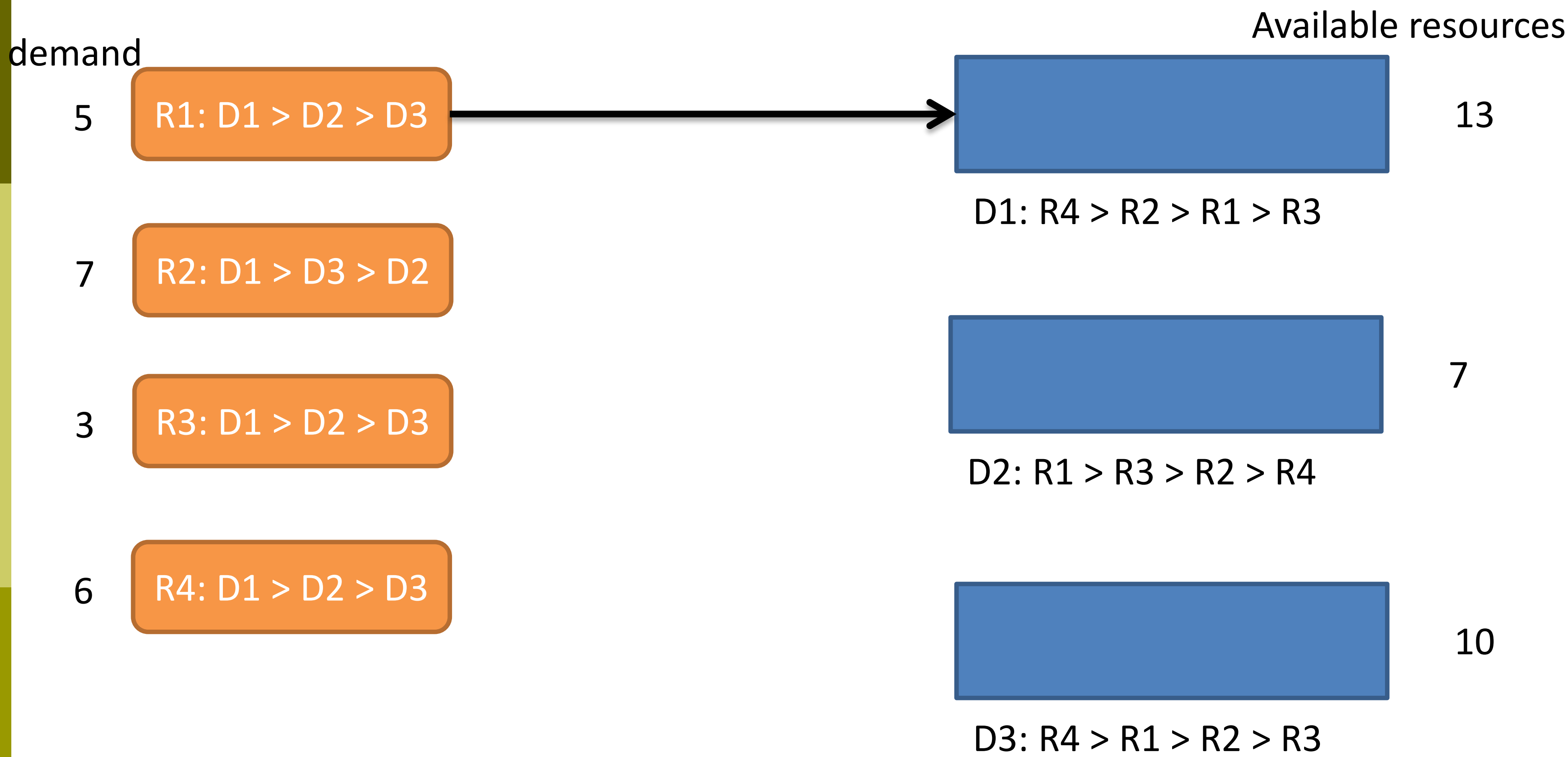
Request R_1
$DC_1 \succ_{R_1} DC_2$
Request R_2
$DC_1 \succ_{R_2} DC_2$
Request R_3
$DC_2 \succ_{R_3} DC_1$
10
q
5
5
6
t
3
3
h
k
Request R_1
$DC_1 \succ_{R_1} DC_2$
6 units of CPU
Request R_2
$DC_1 \succ_{R_2} DC_2$
30 units of CPU
Request R_3
$DC_2 \succ_{R_3} DC_1$
18 units of CPU
φ
48 units of CPU
Data center DC_1
$R_2 \succ_{DC_1} R_1 \succ_{DC_1} R_3$
30 units of CPU
Data center DC_2
$R_2 \succ_{DC_1} R_3 \succ_{DC_1} R_1$
D
C
r
10
12
12
G
F
h
k
6
10
10
Data center DC_1
$R_2 \succ_{DC_1} R_1 \succ_{DC_1} R_3$
Data center DC_2
$R_2 \succ_{DC_1} R_3 \succ_{DC_1} R_1$

SEA vs. Stable Marriage Problem

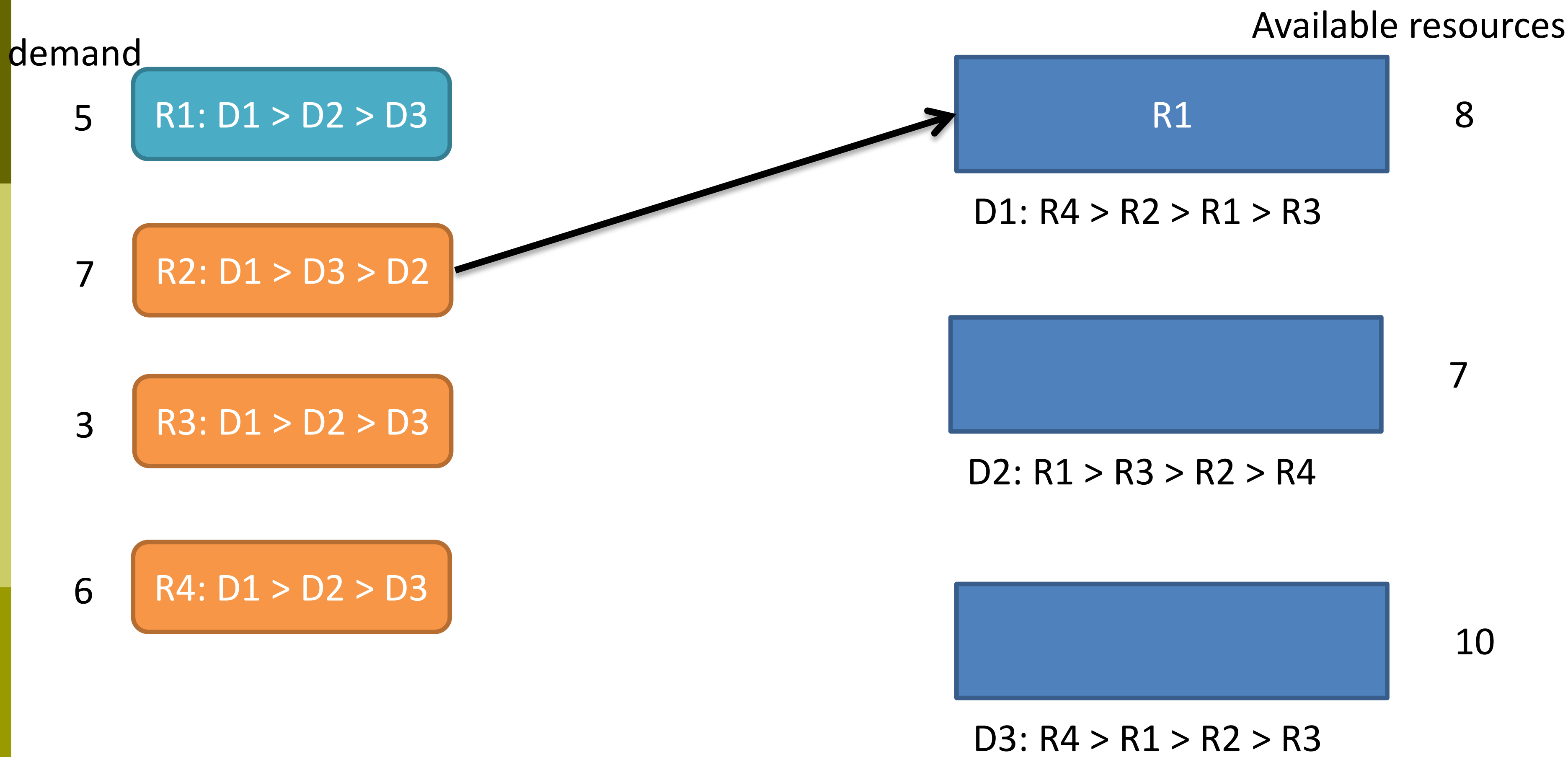
- Stable Marriage Problem (SMP): given n men and n women, each person is associated with a strictly ordered preference list containing all the members of the opposite sex, find a stable matching.

- The differences are
  - Heterogeneous data centers
  - Heterogeneous resource requests

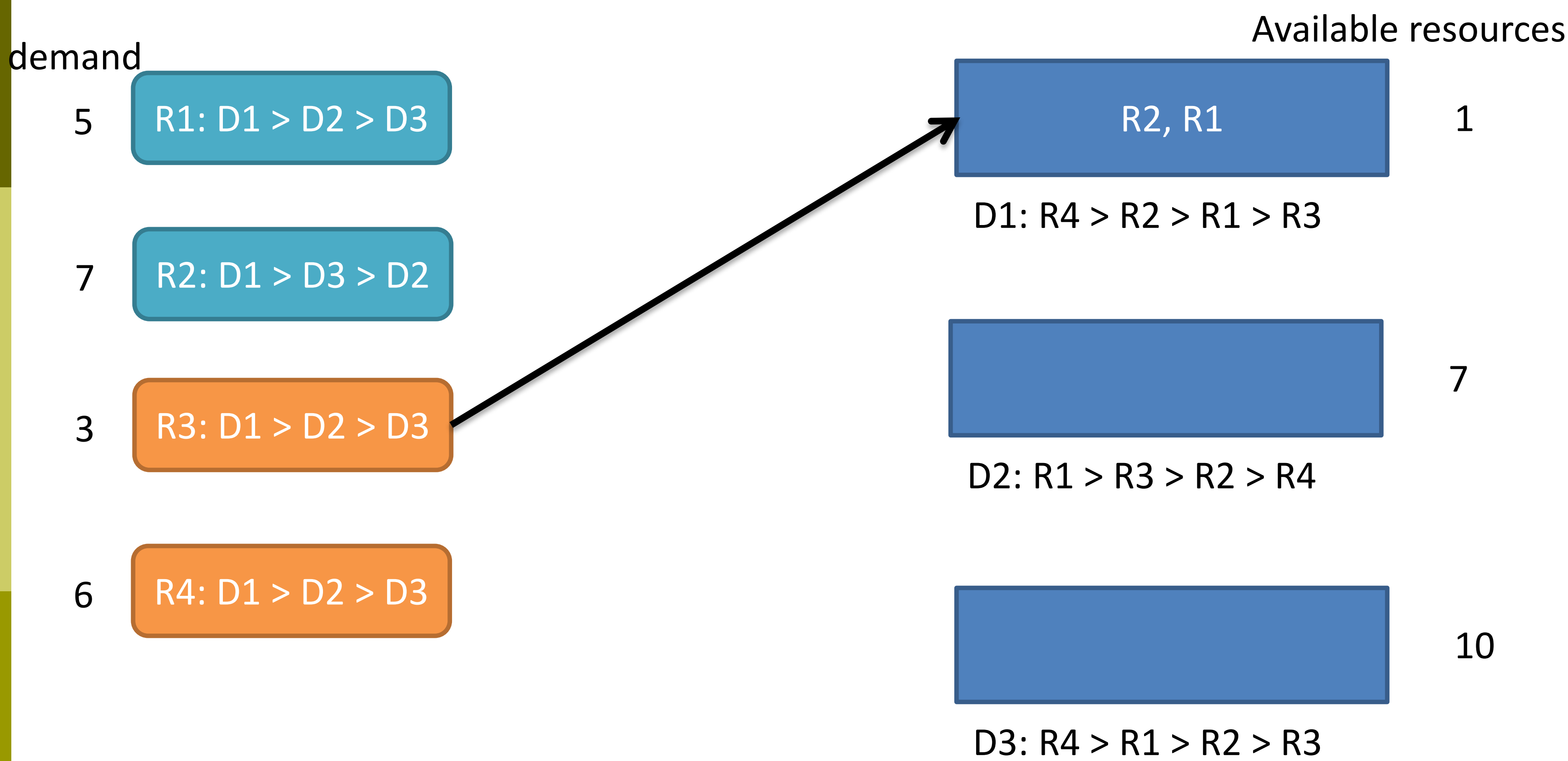
Allocation Algorithm for 1-D SEA (1/10)



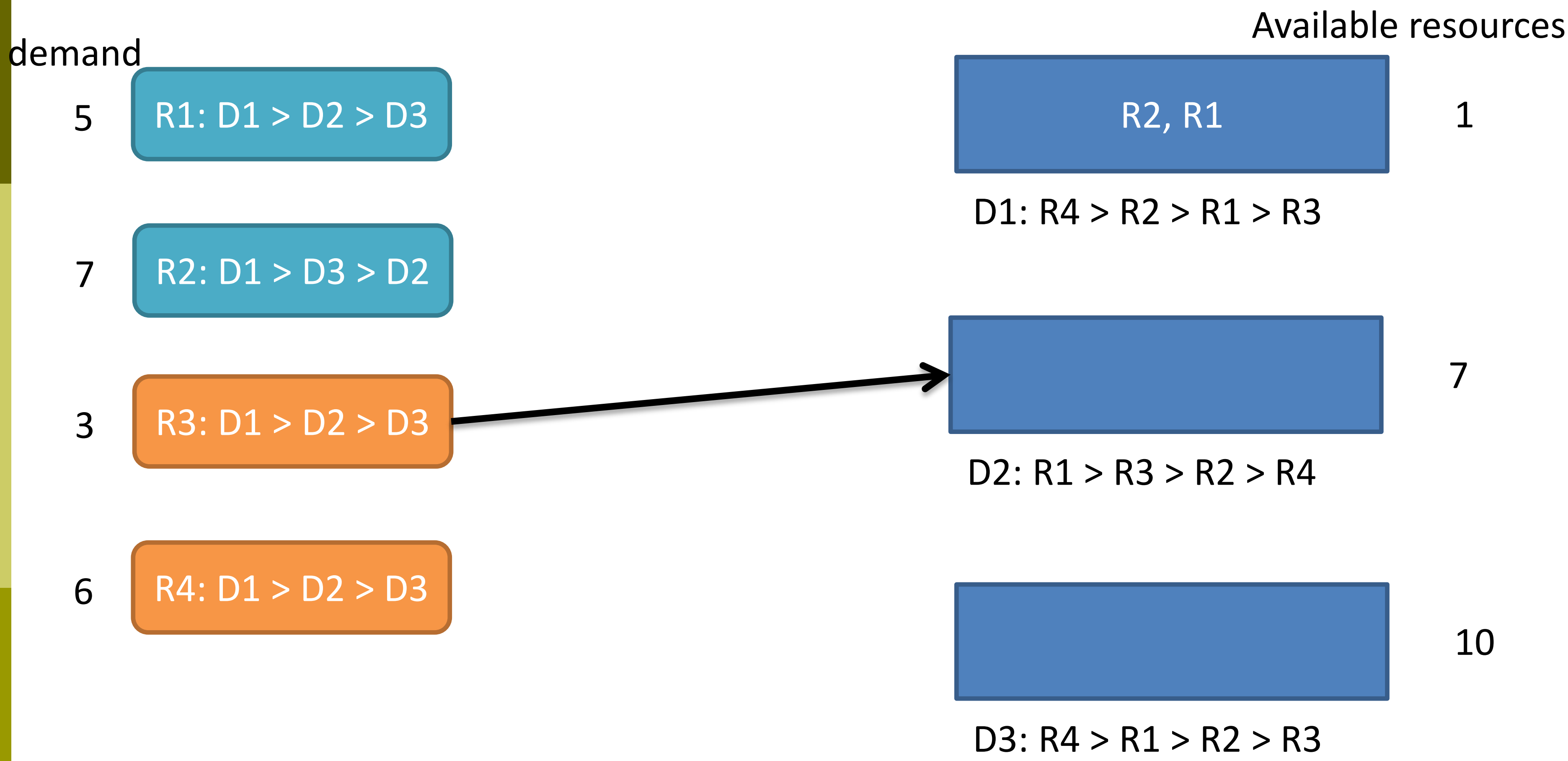
Allocation Algorithm for 1-D SEA (2/10)



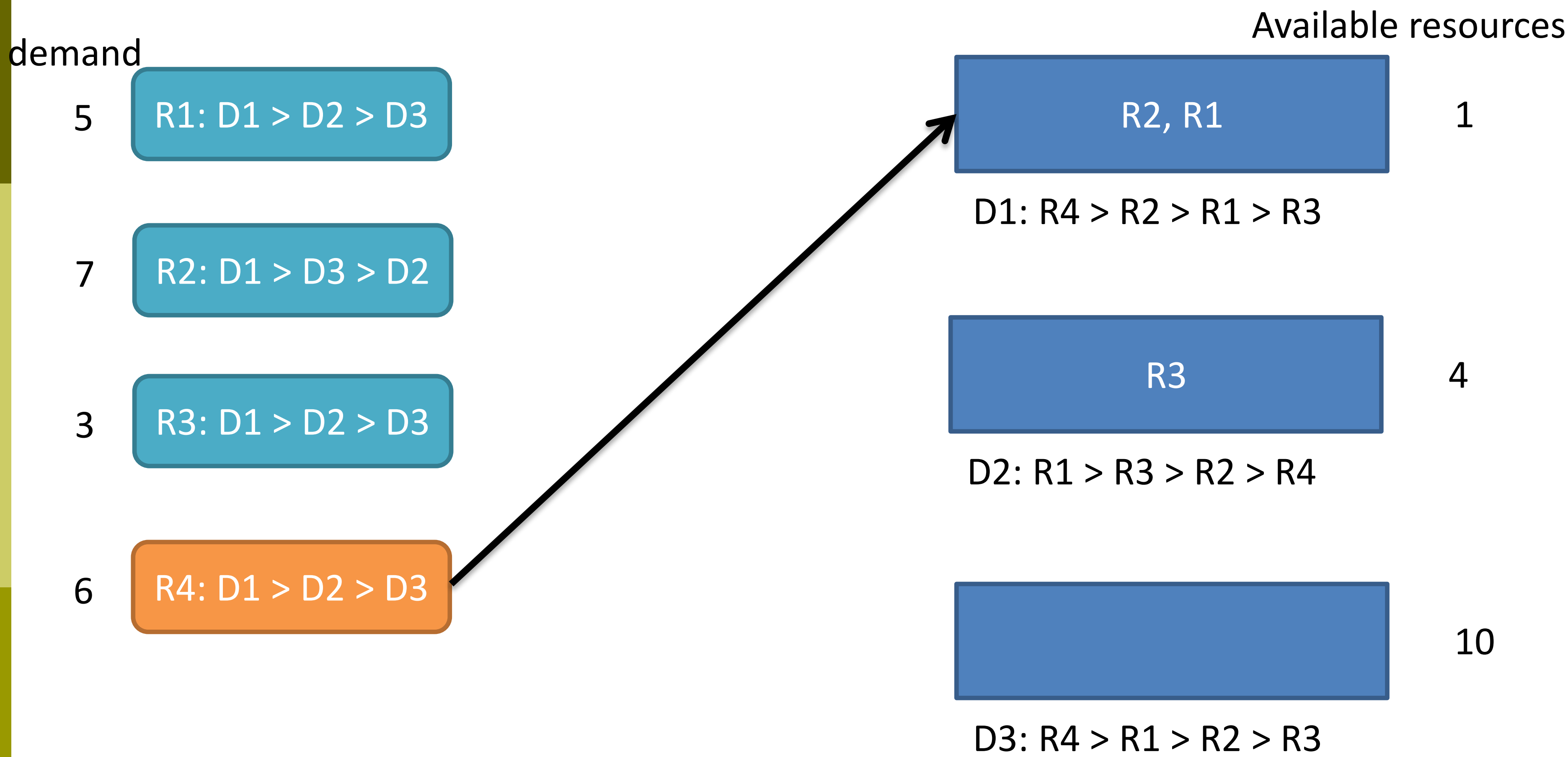
Allocation Algorithm for 1-D SEA (3/10)



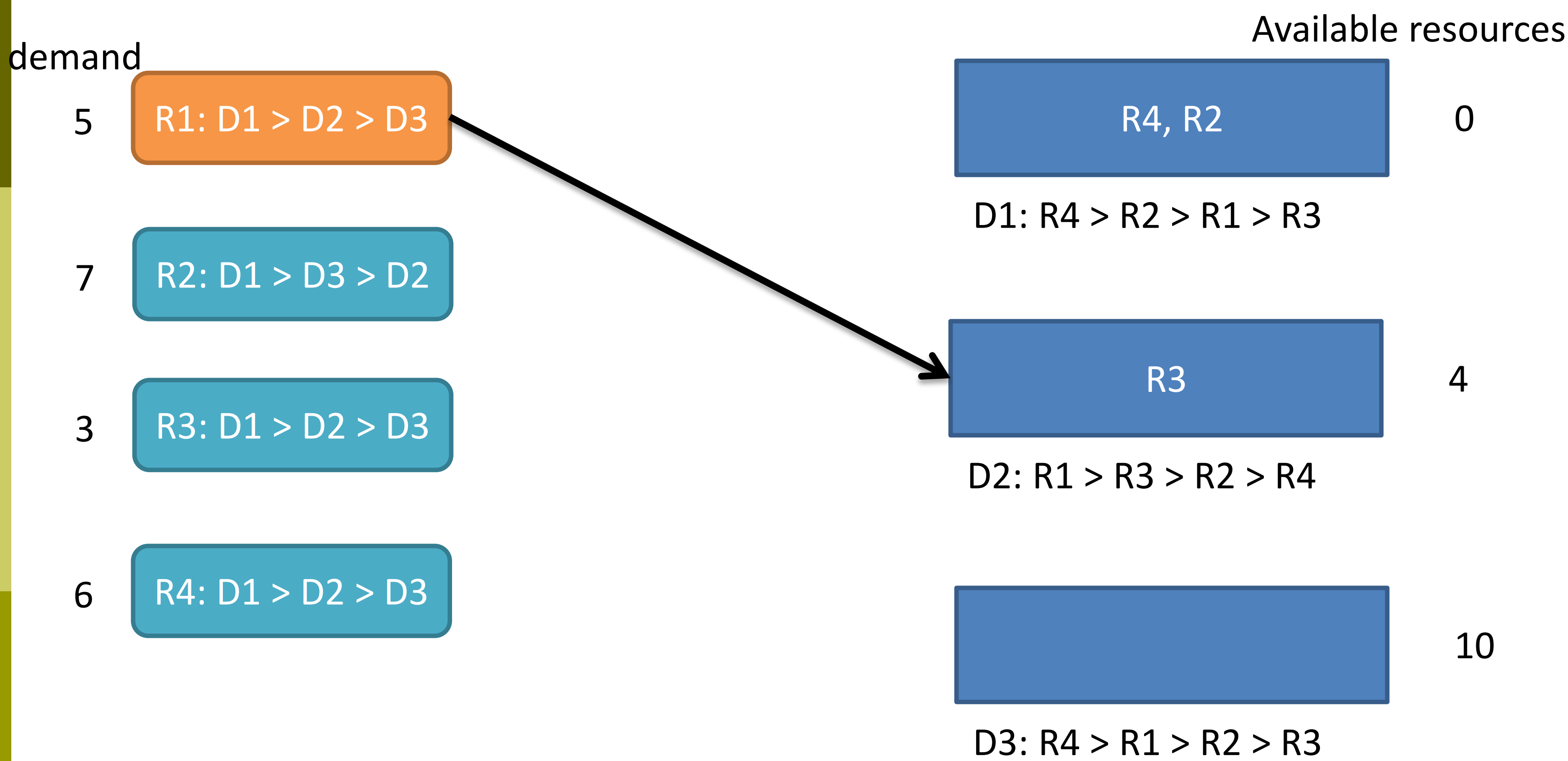
Allocation Algorithm for 1-D SEA (4/10)



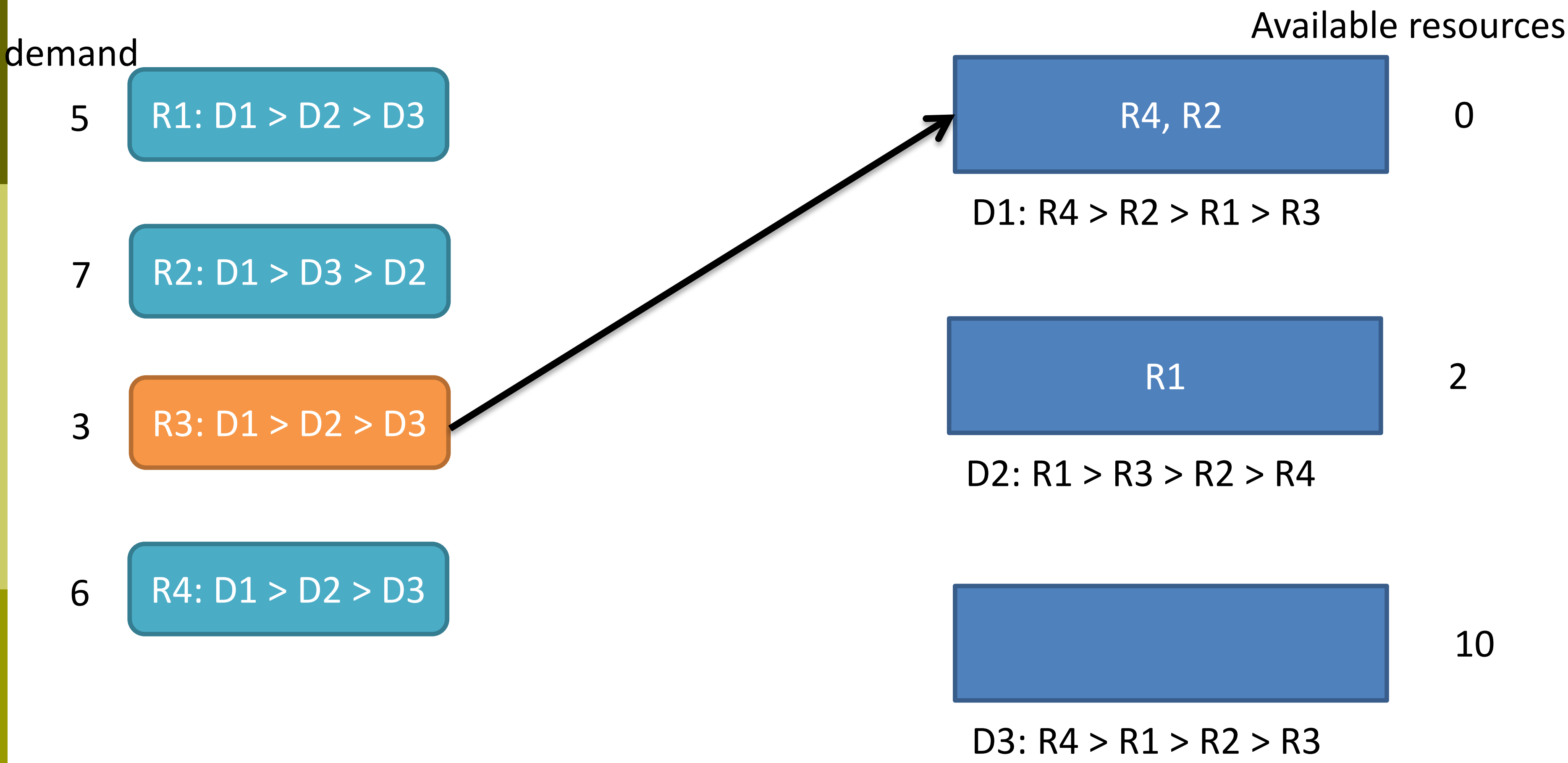
Allocation Algorithm for 1-D SEA (5/10)



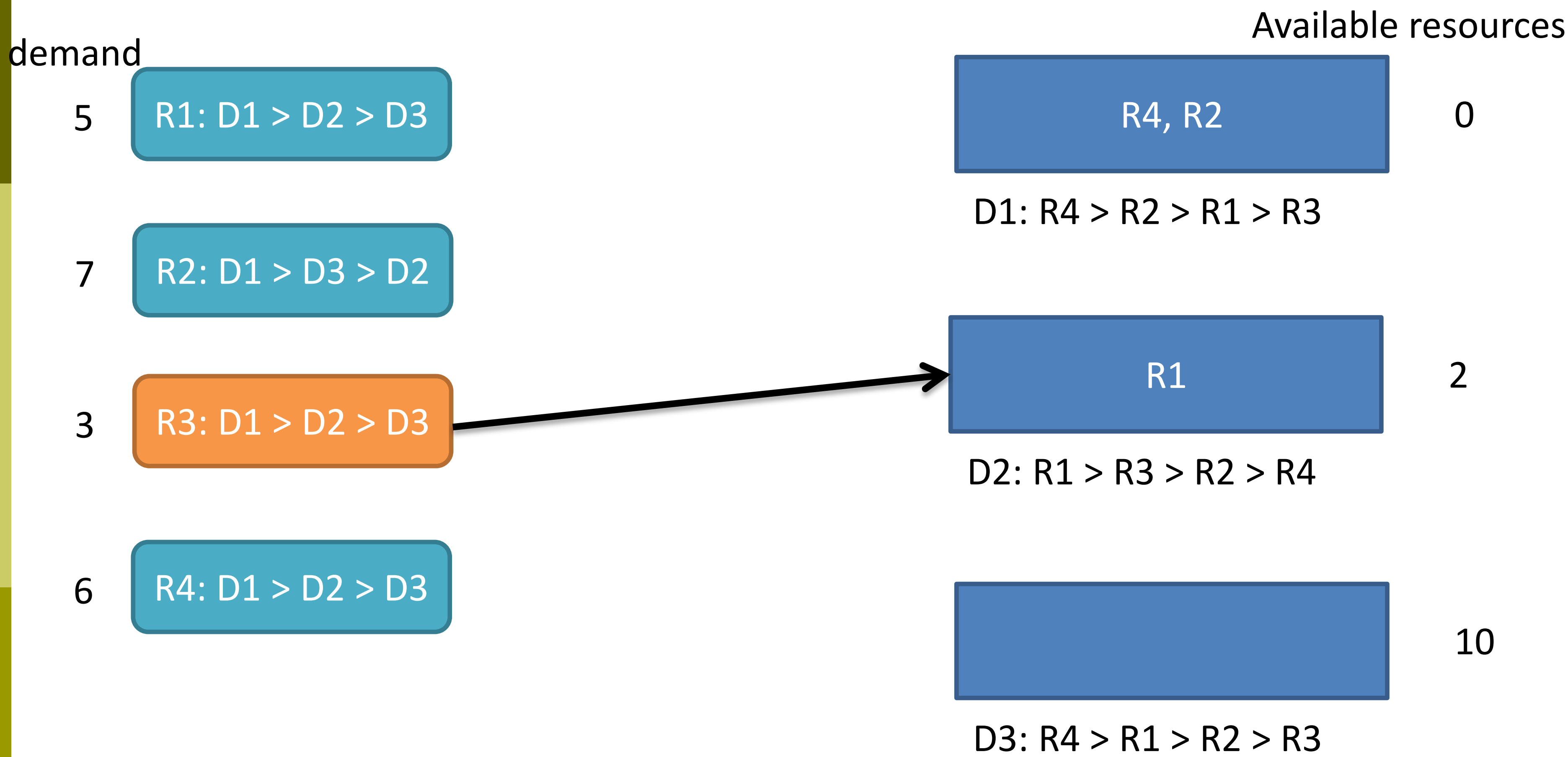
Allocation Algorithm for 1-D SEA (6/10)



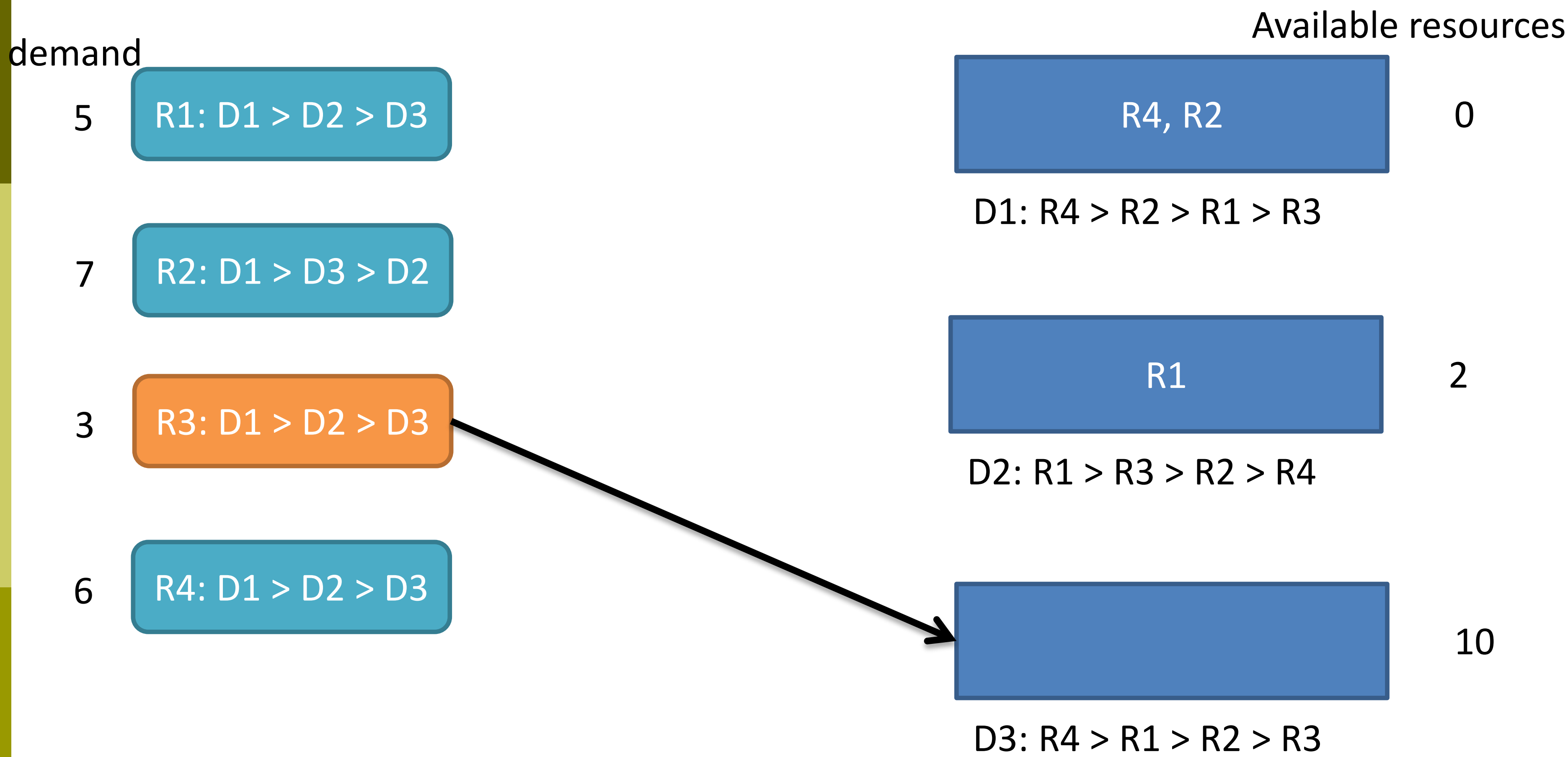
Allocation Algorithm for 1-D SEA (7/10)



Allocation Algorithm for 1-D SEA (8/10)



Allocation Algorithm for 1-D SEA (9/10)



Allocation Algorithm for 1-D SEA (10/10)

demand

5 R1: D1 > D2 > D3

7 R2: D1 > D3 > D2

3 R3: D1 > D2 > D3

6 R4: D1 > D2 > D3

Available resources

R4, R2 — 0

D1: R4 > R2 > R1 > R3

R1 — 2

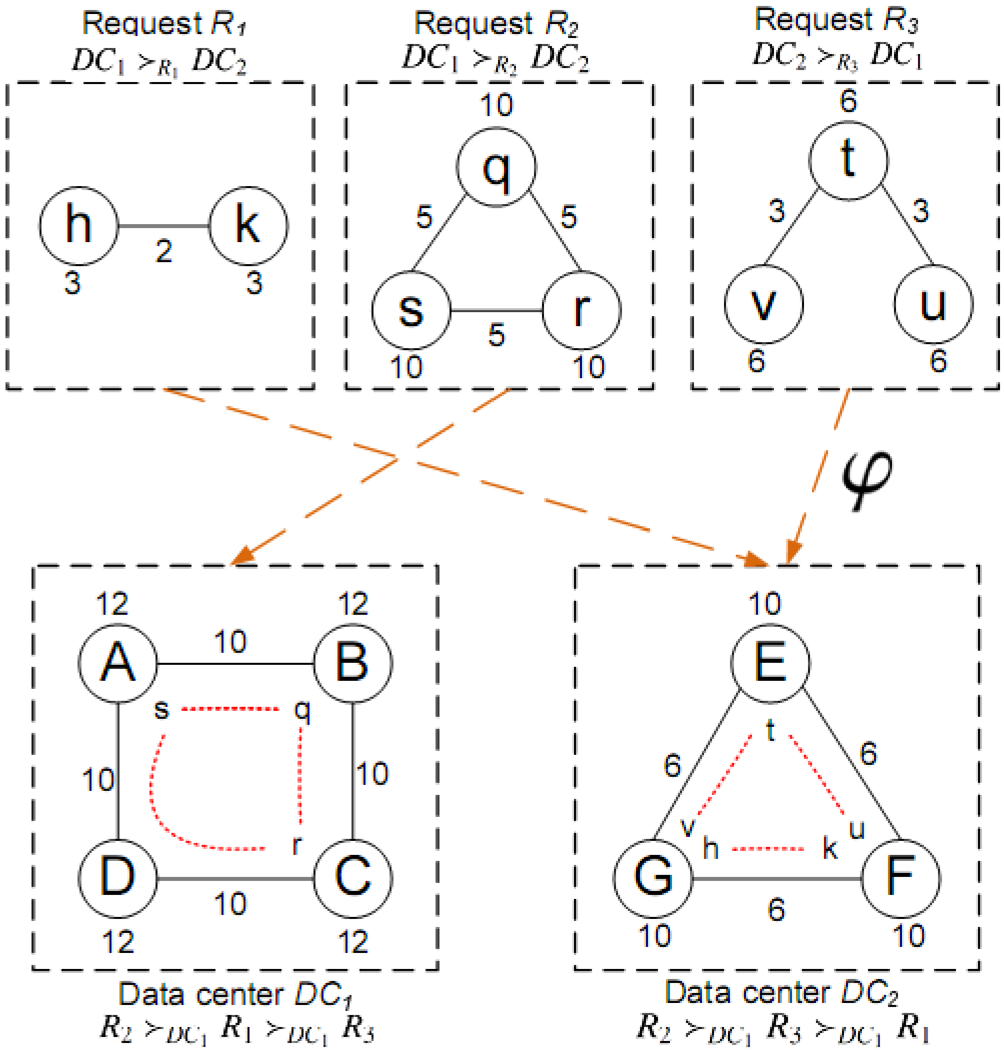
D2: R1 > R3 > R2 > R4

R3 — 7

D3: R4 > R1 > R2 > R3

The allocation is stable, as there is no blocking pair.

2-D SEA

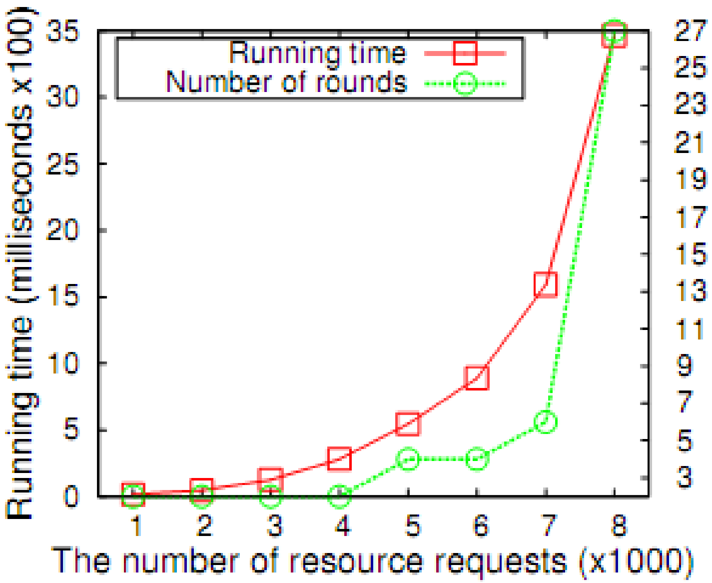


1-D SEA + Virtual Network Embedding

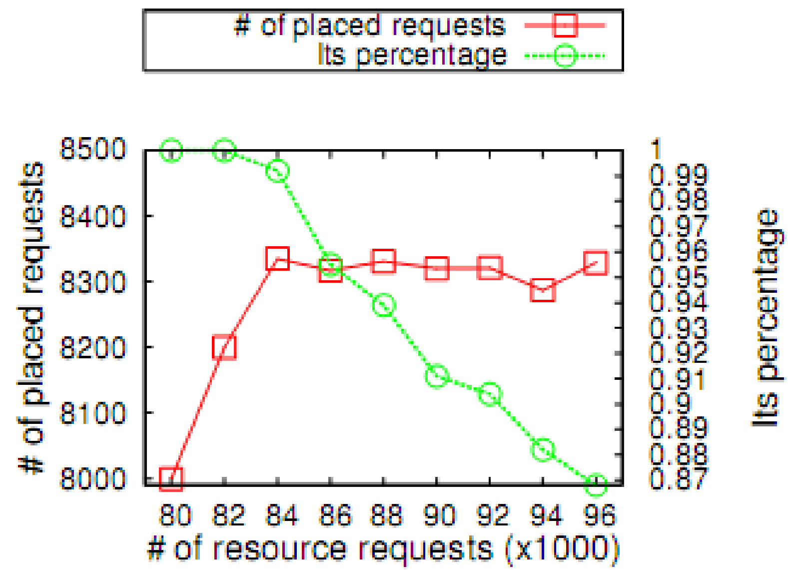
Content

- Stable rEsource Allocation (SEA) Problem
- Solution
- Evaluation
- Conclusions

Simulation Results (1/2)

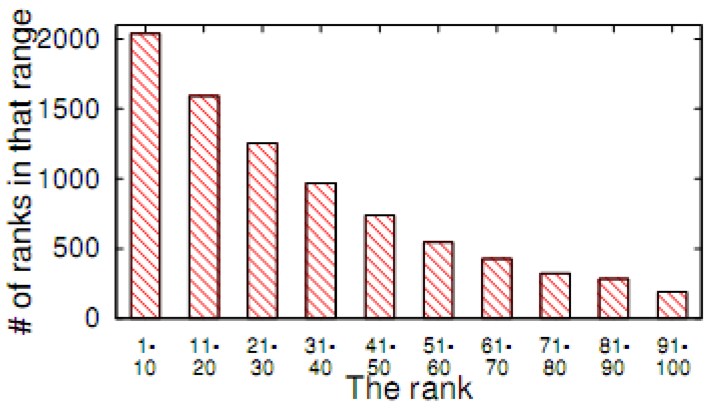


(a) Running time and number of running rounds

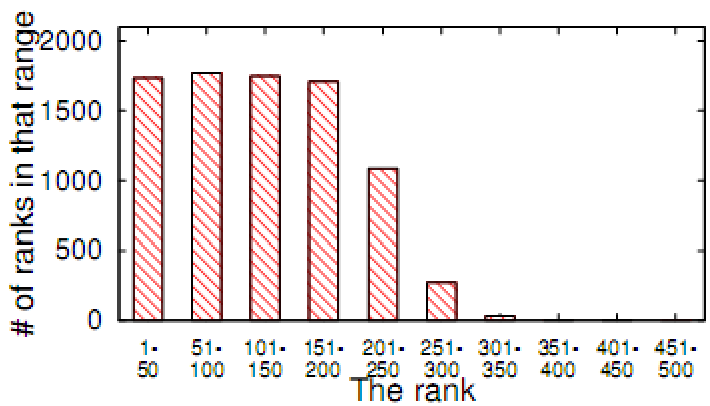


(b) Number of successfully placed requests and its percentage

Simulation Results (2/2)



Quality of allocation from the perspective of requests



Quality of allocation from the perspective of data centers

Content

- Stable rEsource Allocation (SEA) Problem
- Solution
- Evaluation
- Conclusions

Conclusions

- To the best of our knowledge, this is the first attempt that considers stability in resource allocation between multiple providers and tenants.
- We provide a detailed and proper problem formulation.
- We develop efficient and effective solutions for both 1-D and 2-D SEA. Simulation results demonstrate the advantages of both algorithms.

南京大學
NANJING UNIVERSITY

Thanks for your attention!