Resource Optimization for Survivable Embedding of Virtual Clusters in Cloud Data Centers

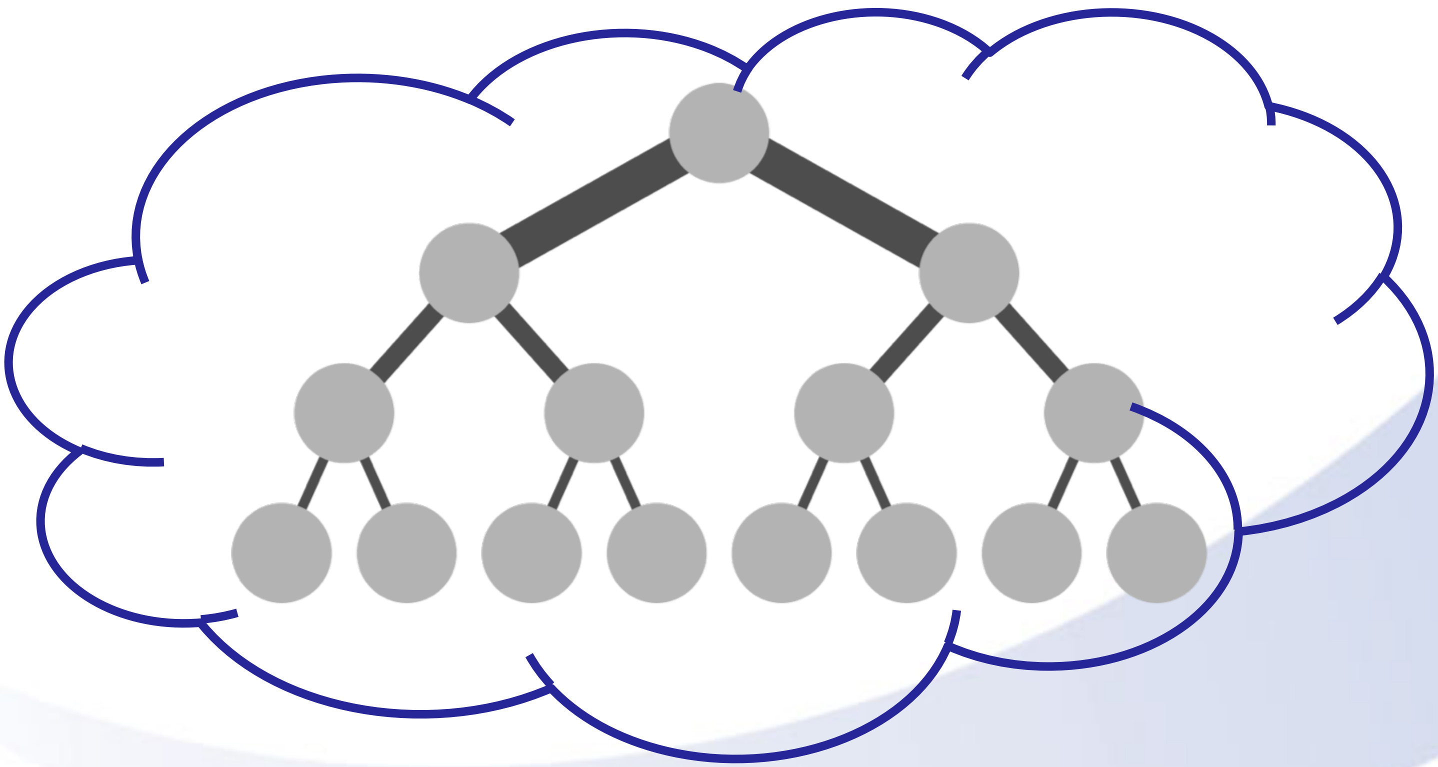
Biyu Zhou[1], Jie Wu[2], Fa Zhang[1], and Zhiyong Liu[1]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]Temple University

Outline

- Backgrounds
- Problem
- Solutions
- Evaluation

Backgrounds

- Virtualization has stimulated the development of cloud computing.



中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

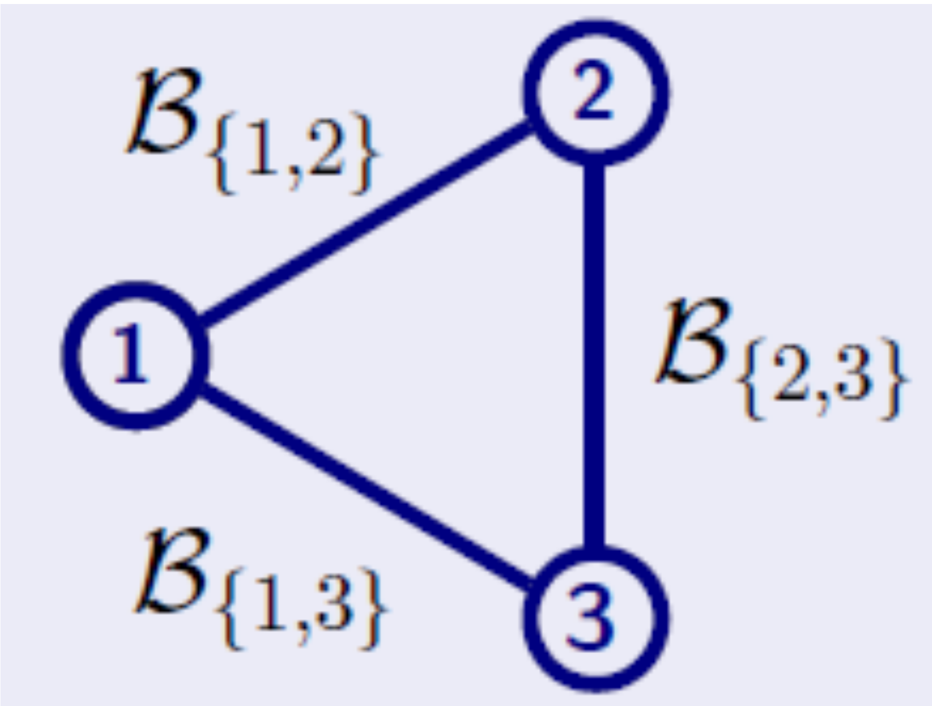
Backgrounds

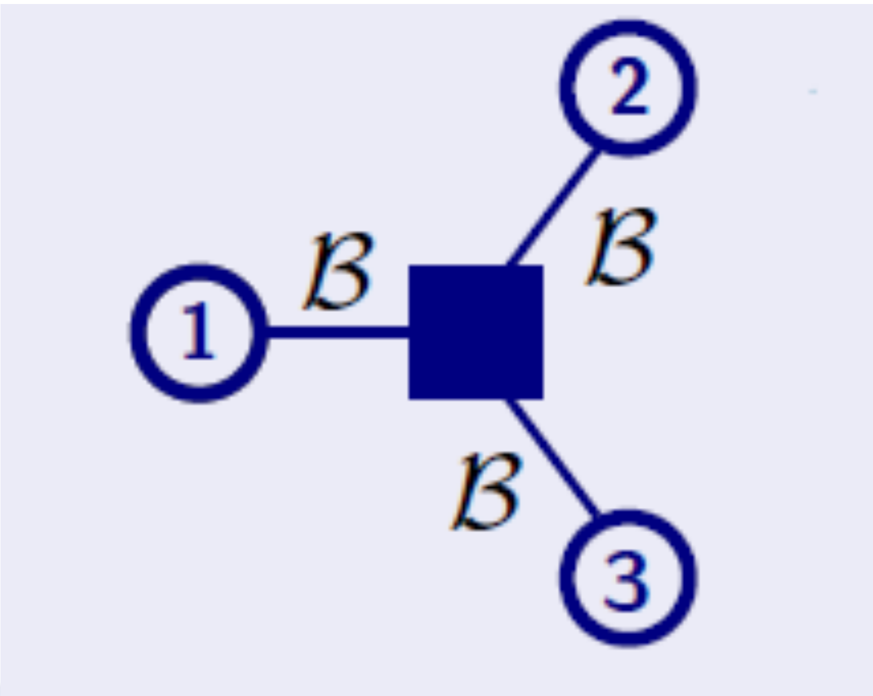
Performance degradation to user experience?

Bandwidth-guaranteed service placement.

Bandwidth-guaranteed VM allocation

- Early 2000s: Graph Abstraction Embedding (GAE)
- Since 2011: Virtual Cluster Embedding (VCE)





Backgrounds

Physical server failures happen frequently in large scale cloud datacenters?

Survivable Virtual Cluster Embedding!

Survivable VCE

- Proactive survival mechanism
  - Some techniques can provide an advanced warning of expected hardware failure.
  - Providing extra VM slots and bandwidth in case of failure!

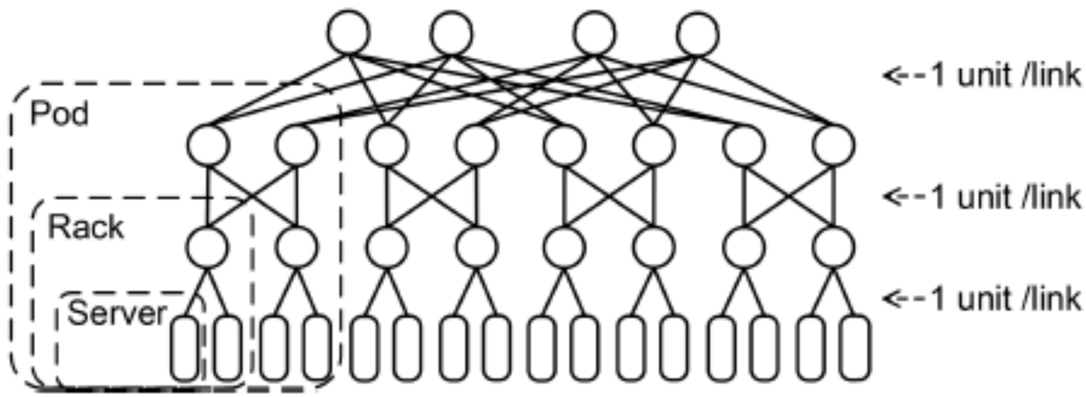
- 1-survivability constraint
  - While single-server failures are frequent, simultaneous multi-server failures are rare.
  - Service can be recovered in the event of an arbitrary single physical server failure.

Outline

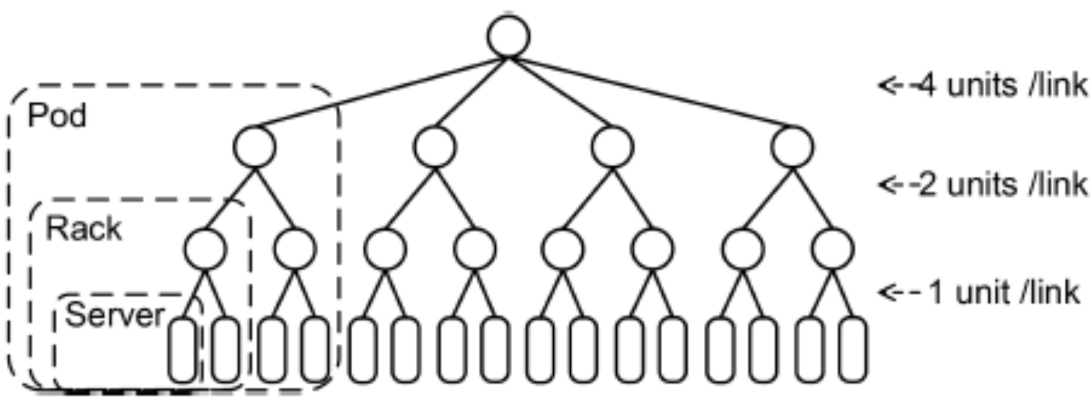
- Background
- Problem
- Solutions
- Evaluation

Data center topology

- A typical type of DCN topology: Tree-like structure.



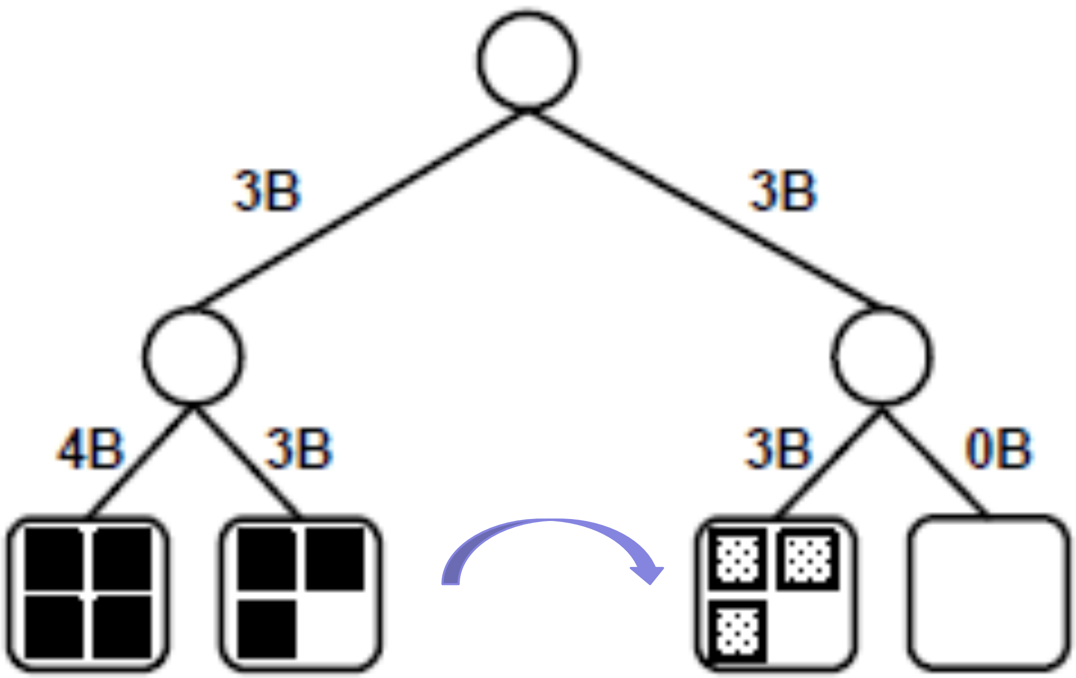
(a) Original 4-ary FatTree topology.



(b) Equivalent abstraction of a 4-ary FatTree topology.

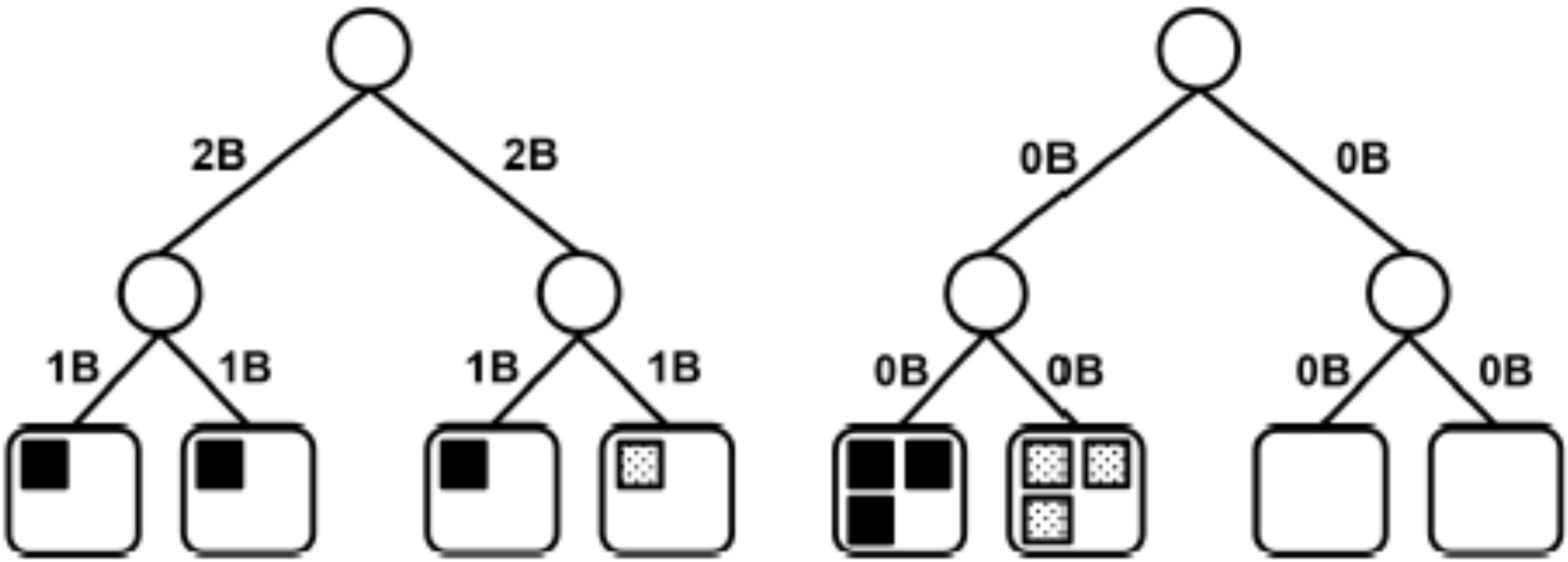
Survivable VCE

- Primary VMs: active during normal operations
- Backup VMs: in standby mode, activated when a primary VM’s PM fails
  - Each backup VM synchronizes the states of multiple primary VMs



Resource minimization

- Server resource: the average number of VM slots reserved.
- Bandwidth resource: the average bandwidth units reserved.



(a) (b)

Abstracting the problem

- Problem
  - Can we find an embedding plan for all the requests in the network such that total resource cost is minimized while the constraints on either survivability or resource capacity are not violated?

- Solving the problem is NP-hard!

Outline

- Background
- Problem
- Solutions
- Evaluation

Solution for single VC request

- Observation 1: minimizing the server resource usage is equivalent to minimize the maximum number of primary VMs in a single physical server.

- Observation 2: minimizing the bandwidth resource usage can be achieved by packing VMs together as much as possible.

Solution for single VC request

- Nb: the maximum number of primary VMs in a single physical server

Start

Nb=1

Nb=2

…

Nb=min{C,N}

Find the embedding that minimize the total resource

End

- C: the server capacity
- N: the number of VMs in a request

Solution for single VC request

- Question: how to find the embedding with minimum bandwidth resource?

- Observation: minimizing the bandwidth resource usage can be achieved by packing VMs together as much as possible.

- Method: search feasible embedding plans from bottom to up, use the one with the lowest level.

Solution for multiple VC requests

- Observation: the embedding of the VC request with a larger weight has greater impact on the final performance.

- Solution:
  - Sort all the VC requests in descending order of their weights.
  - All the VC are embedded one-by-one according to the solution for single VC request.

Outline

- Background
- Problem
- Solutions
- Evaluation

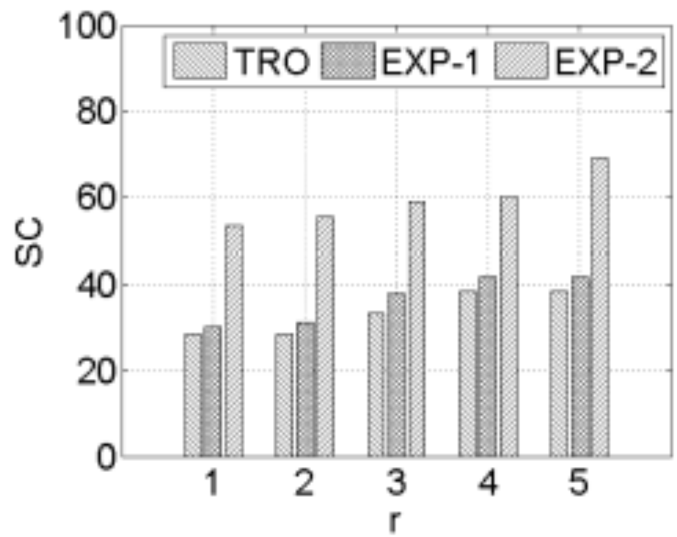
The evaluation setting

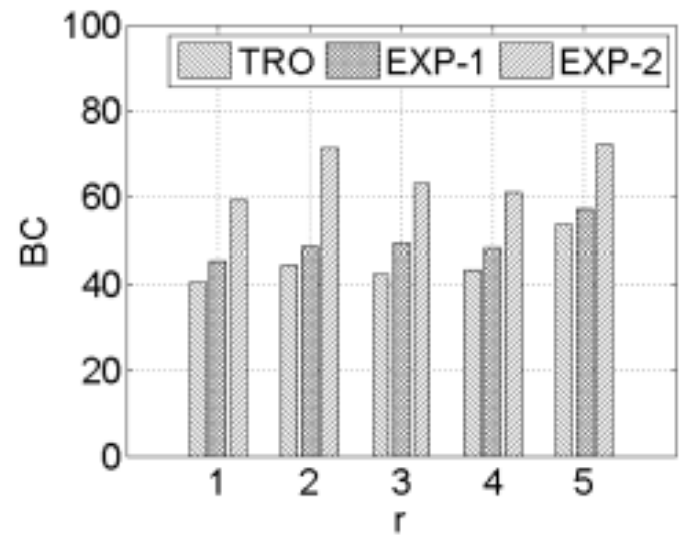
- Benchmarks
  - Our algorithm (TRO)
  - Shadow-based algorithm (EXP-1)
  - Minimal VM slots algorithm (EXP-2)

- Evaluation metric
  - Total resource consumption (TC).
  - Server resource consumption (SC).
  - Bandwidth resource consumption (BC).

Single VC request scenario

α	TC	SC	BC
0	34.5	18.5	16
0.25	33.5	16	17.5
0.5	34.5	15	19.5
0.75	37.5	14.5	23
1	38.5	13	25.5

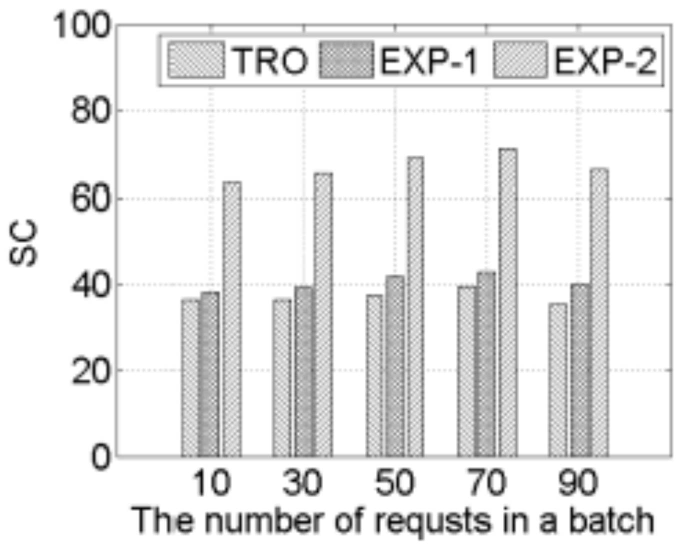


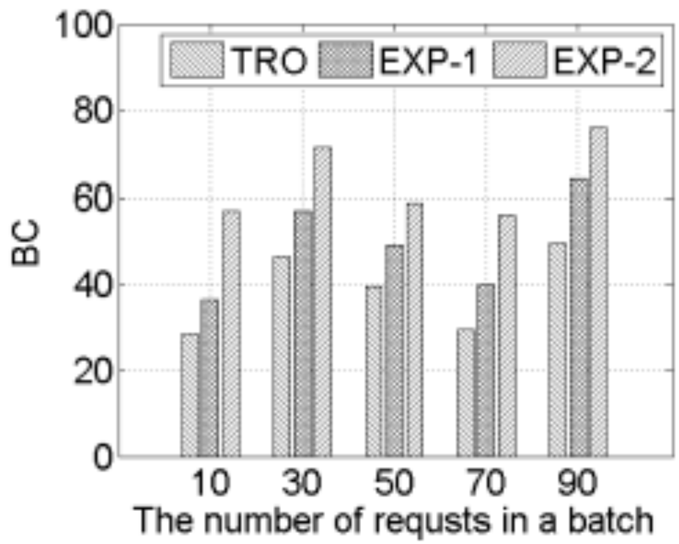


- (a) By changing the value of α, we can control the trade-off between SC and BC.
- (b) TRO algorithm outperforms the newly proposed EXP-2 algorithm in terms of bandwidth resource consumption (i.e., BC) while achieving a similar performance in terms of server resource consumption (i.e., SC).
- (c) EXP-2 works better than EXP-1.

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

Multiple VC requests scenario





- (a) TRO outperforms EXP-2 and is close to EXP-1 in terms of SC.
- (b) TRO outperforms EXP-1 and EXP-2 in terms of bandwidth resource consumption.
- (c) TRO has a better request serving capacity in the batch scenario.

Thanks for your attention!

zhoubiyu@ict.ac.cn