Bandwidth Guarantee under Demand Uncertainty in Multi-tenant Clouds

Lei Yu, Haiying Shen

Clemson University

ICDCS 14

Outline

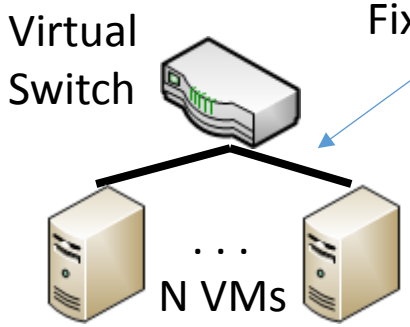
- Introduction
- Stochastic cloud network sharing
- VM allocation algorithms
- Evaluation
- Conclusion and future work

Introduction

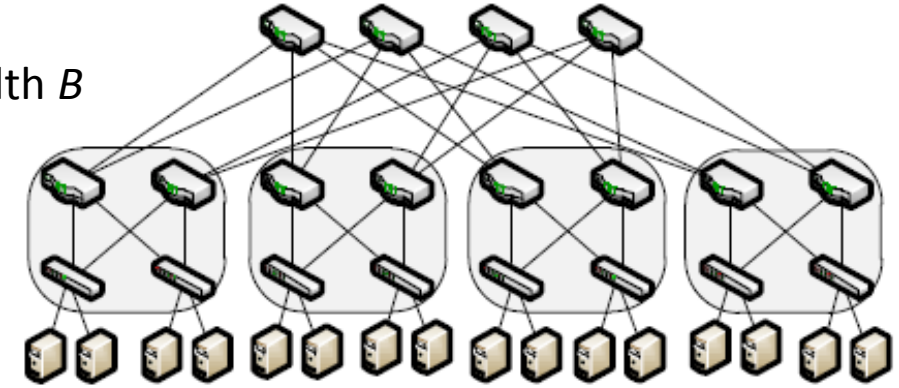
- Cloud Computing
  - Infrastructure: shared, multi-tenant datacenters
  - Contention: Applications compete for the scarce network resources
  - Performance Variability: contention and lack of guaranteed network bandwidth lead to variable data transmission latency and job completion time

Bandwidth Reservation in Data centers

- SecondNet (Co-NEXT‘10)
  - end-to-end bandwidth reservation per VM-pair
- Oktopus (Sigcomm’11)
  - virtual cluster



Virtual Cluster Model



Production Data Center

1. Determine the model <*N,B*>

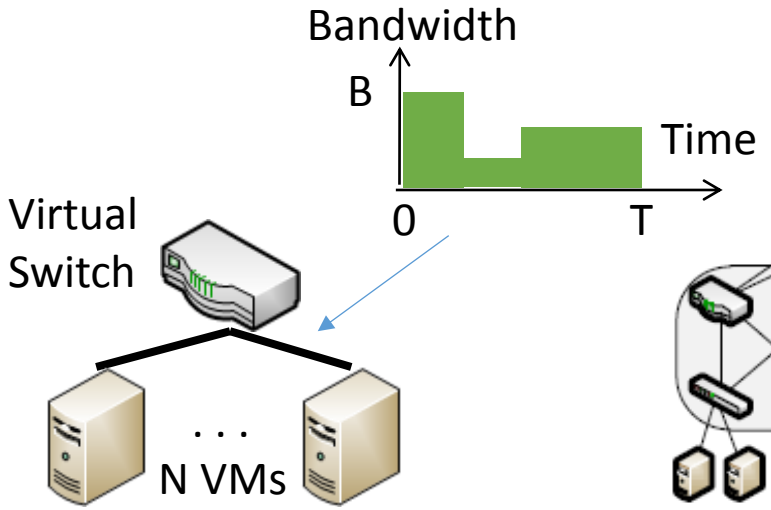
2. Allocate and enforce the model

Bandwidth Reservation in Data centers

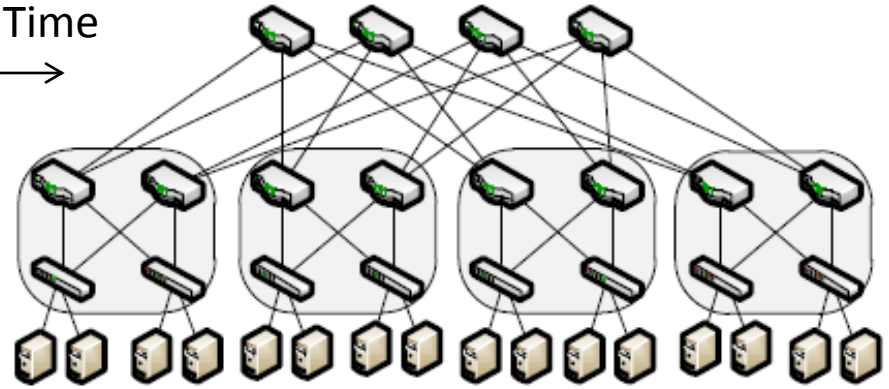
- SecondNet (Co-NEXT’ 10)
  - End-to-end bandwidth reservation per VM-pair
- Oktopus (Sigcomm’11)
  - Virtual cluster
- TIVC (Sigcomm’12)
  - Time-varying BW reservations

Bandwidth Reservation in Data centers

- SecondNet (Co-NEXT 10’)
  - End-to-end bandwidth reservation per VM-pair
- Oktopus (Sigcomm 11’)
  - Virtual cluster
- TIVC (Sigcomm 12’)



Virtual Cluster Model



Production Data Center

1. Determine the model <*N,B(t)*>

2. Allocate and enforce the model

Challenges

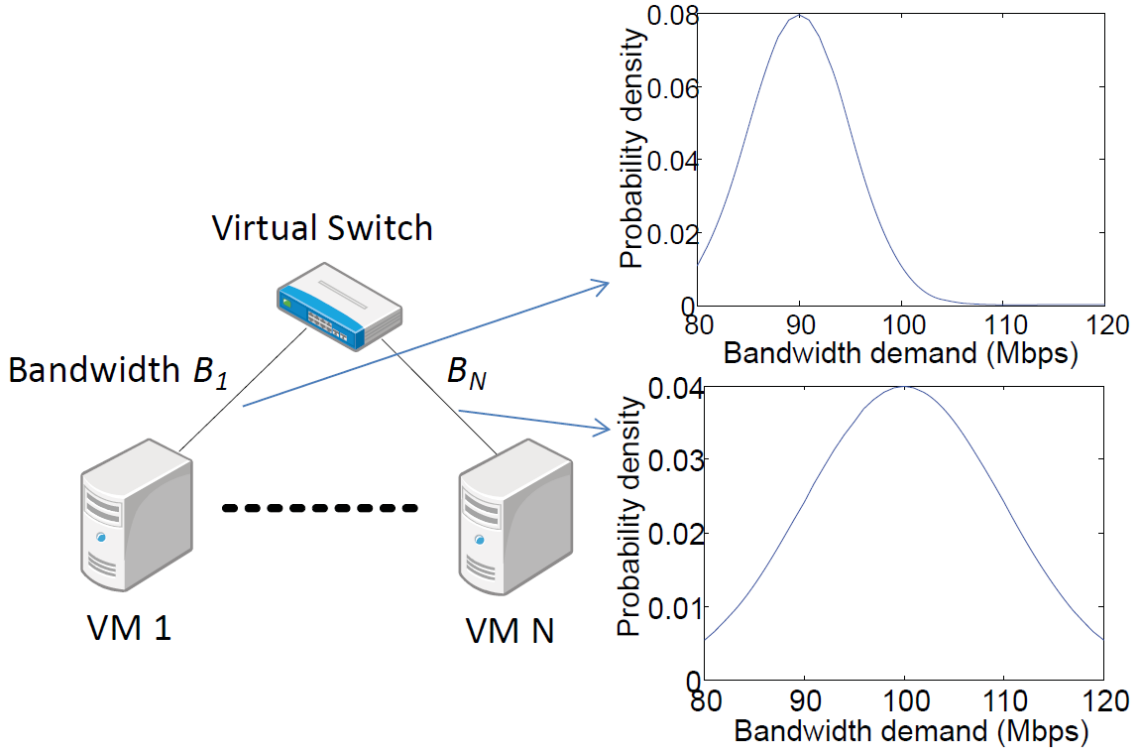
- Existing approaches require reliable and deterministic estimate of bandwidth demand.
- However, the network traffic is highly volatile in production datacenters. It is difficult for a tenant to determine the exact amount of bandwidth it needs at a particular time under demand uncertainty.

Stochastic cloud network sharing

- Stochastic Virtual Cluster (SVC) Model
  - $< N, B_1, B_2, \ldots, B_N >$
  - B_i is random variable
  - In this paper, we assume normal distribution, that is, $< N, (\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_N, \sigma_N) >$, where μ_i is the mean and σ_i^2 is the variance.

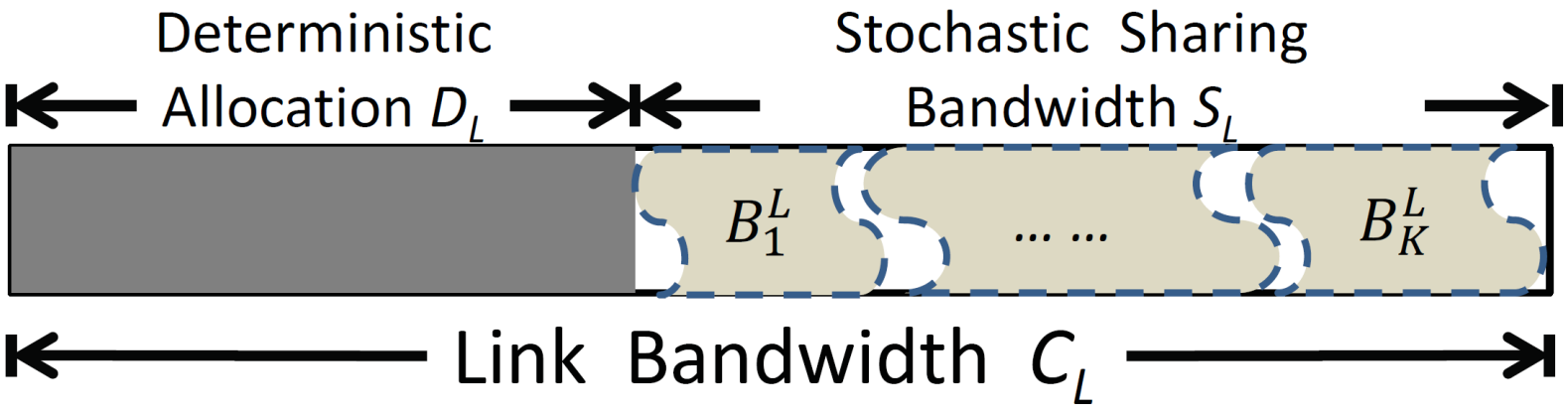
$if\ \mu_1 = \mu_2 = \ldots = \mu_N$ and $\sigma_i = 0, \forall i,$ SVC

is equal to deterministic virtual cluster in Oktopus .



Stochastic cloud network sharing

- Probabilistic Bandwidth Guarantee



$$\Pr\left(\sum_{i=1}^{K} B_i^L > S_L\right) < \varepsilon$$

B_i^L is random bandwidth demand of K virtual clusters on link L

This inequality describes that the bandwidth outage on link L is only allowed to happen with a small probability ε

VM allocation problem

- For the tenant's request $< N, (\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_N, \sigma_N) >$, to allocate *N* empty VM slots such that each physical link *L* can still achieve the probabilistic guarantee for all stochastic bandwidth demands *L* carries, i.e.,

$$\Pr\left(\sum_{i=1}^{K} B_i^L > S_L\right) < \varepsilon$$

VM allocation for homogeneous bandwidth demand

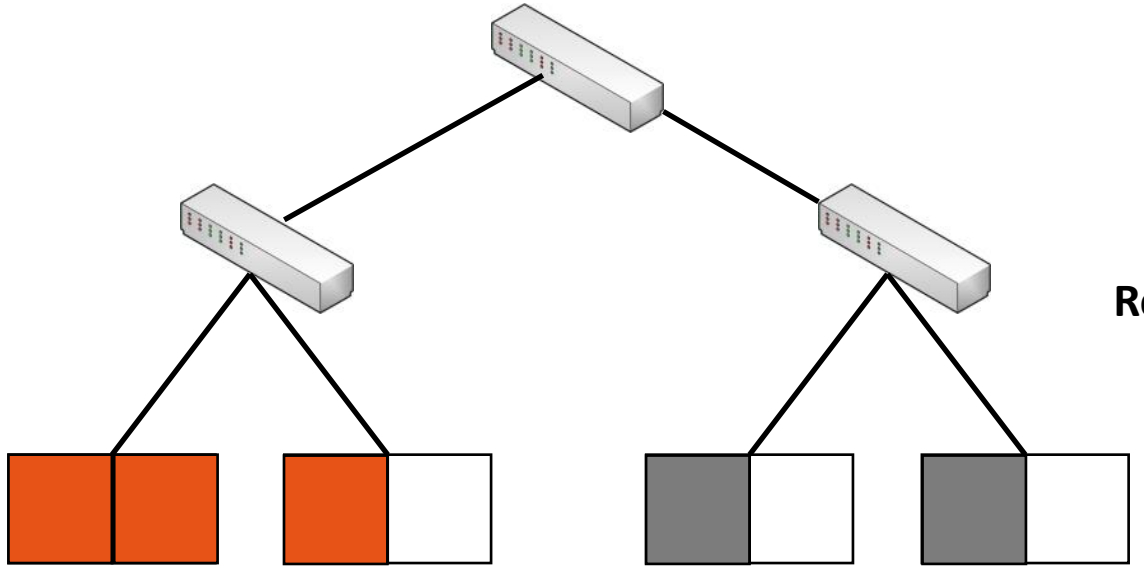
- $< N, (\mu_1, \sigma_1), (\mu_2, \sigma_2), \ldots, (\mu_N, \sigma_N) >$

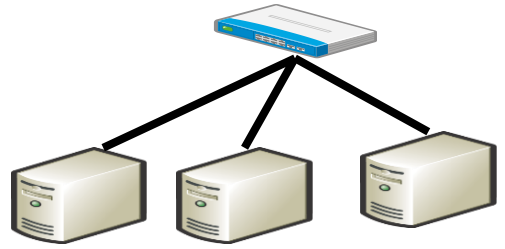
homogeneous bandwidth demand: $\mu_i = \mu, \sigma_i = \sigma, \forall i$.

< *N*, μ, σ >

Allocation algorithm

- Tree network topology
- Allocate a SVC to a subtree, in which there are enough empty VM slots and any link *L* can satisfy the bandwidth requirement

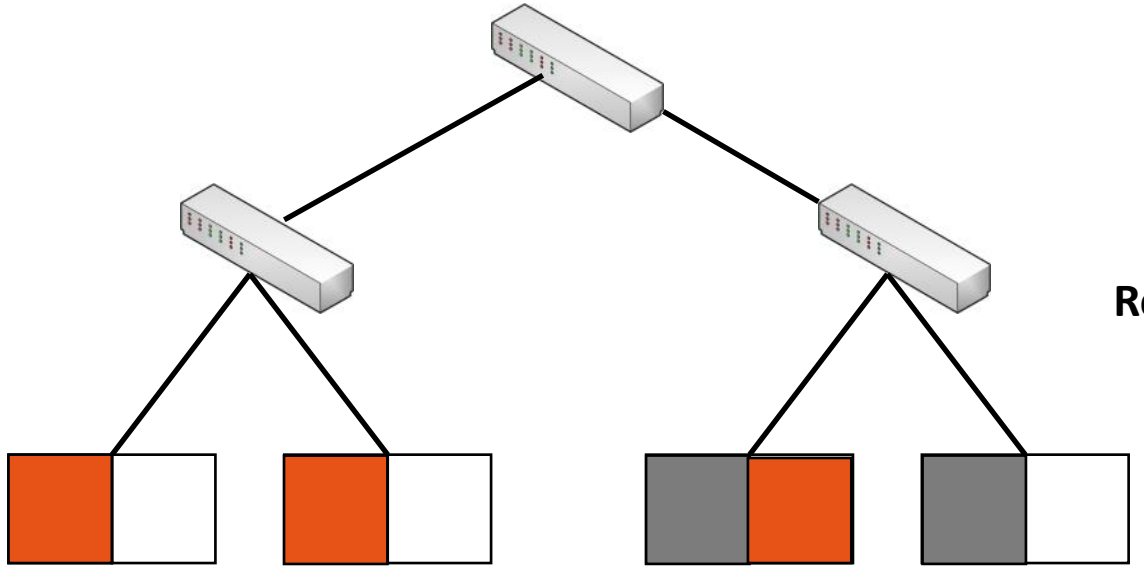


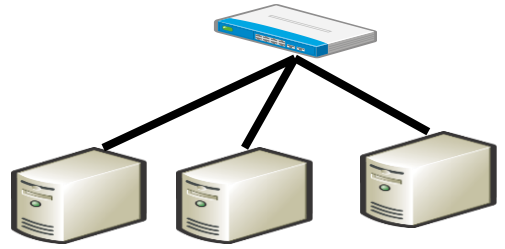


Request : <3 VMs, 100 Mbps, 20 >

Allocation algorithm

- Tree network topology
- Allocate a SVC to a subtree, in which there are enough empty VM slots and any link *L* can satisfy the bandwidth requirement





Request : <3 VMs, 100 Mbps, 20 >

How to find valid allocation?

VM allocation

- Oktopus: for a deterministic virtual cluster <***N,B***>, bandwidth needed on a link that connects ***m*** VMs to the remaining ***(N-m)*** VMs is = ***Min (m, N-m) * B***

- SVC: for SVC < N, μ, σ >, the aggregate bandwidth demand of m VMs $\mathbb{B}(m)$, following normal distribution $N(m\mu, m\sigma^2)$. Accordingly, the bandwidth needed on a link that connects ***m*** VMs to the remaining ***(N-m)*** VMs is random variable $Min(\mathbb{B}(m), \mathbb{B}(N-m))$.

VM allocation

- Valid allocation condition
  - To satisfy $\Pr\left(\sum_{i=1}^{K} B_i^L > S_L\right) < \varepsilon$ $\quad B_i^L \sim N(\mu_{i,L}, \sigma_{i,L}^2)$

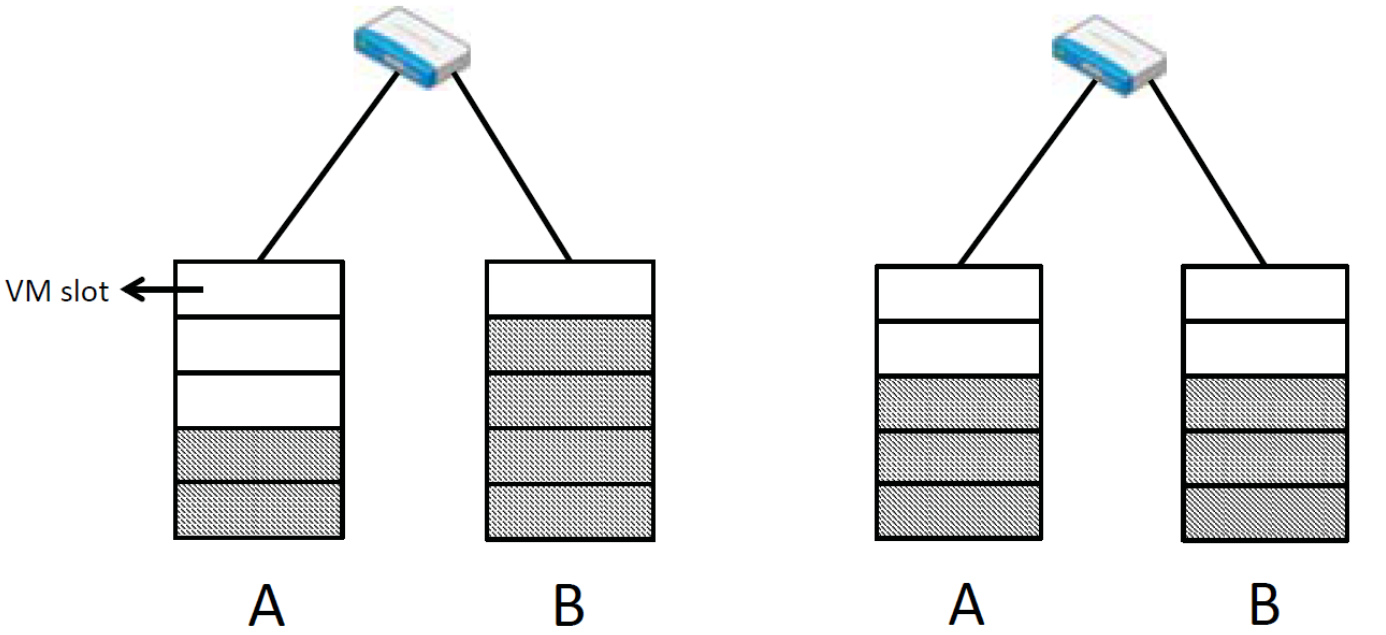
By using normal distribution to approximate the distribution of $\sum_{i=1}^{K} B_i^L$ (central limit theorem), we have

$$\frac{S_L - \sum_{i=1}^{K} \mu_{i,L}}{\sqrt{\sum_{i=1}^{K} \sigma_{i,L}^2}} > \Phi^{-1}(1-\varepsilon)$$

Accordingly, given a VM allocation solution for a SVC request $r = \langle N, \mu, \sigma \rangle$ that allocates m VMs and $N - m$ VMs in two network components divided by any link L,
(1) check whether the two components have no less than m and $N - m$ empty slots in their physical machines respectively,
(2) compute the corresponding $u_{r,L}$ and $\sigma_{r,L}$ and then check whether such allocation is valid by examining the above condition.

Allocation algorithm

- Previous algorithm for TIVC (Sigcomm' 12)



(a) 2 VMs in A, 4 VMs in B (b) 3 VMs in A, 3 VMs in B

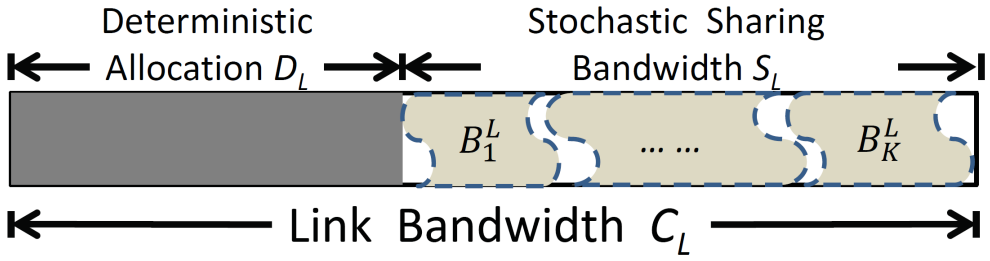
Both allocations are valid, but (a) has lower bandwidth occupancy.

Previous algorithm only finds a valid allocation without optimizing the bandwidth occupancy.

Allocation algorithm

- Bandwidth occupancy

$$S_L = C_L - D_L$$



$$\frac{S_L - \sum_{i=1}^{K} \mu_{i,L}}{\sqrt{\sum_{i=1}^{K} \sigma_{i,L}^2}} > \Phi^{-1}(1-\varepsilon)$$

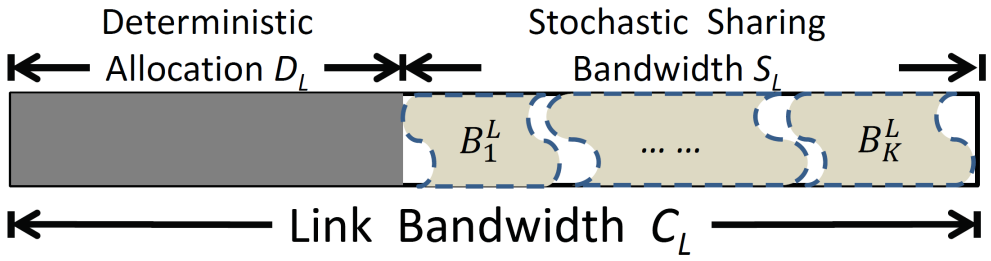
$$S_L > \sum_{i=1}^{K} \mu_{i,L} + \Phi^{-1}(1-\varepsilon)\sqrt{\sum_{i=1}^{K} \sigma_{i,L}^2} = \sum_{i=1}^{K}\left(\mu_{i,L} + \Phi^{-1}(1-\varepsilon)\frac{\sigma_{i,L}^2}{\sqrt{\sum_{i=1}^{K} \sigma_{i,L}^2}}\right)$$

effective amount of bandwidth reserved for *i*-th stochastic bandwidth demand B_i^L , denoted by E_i^L

Allocation algorithm

- Bandwidth occupancy ratio

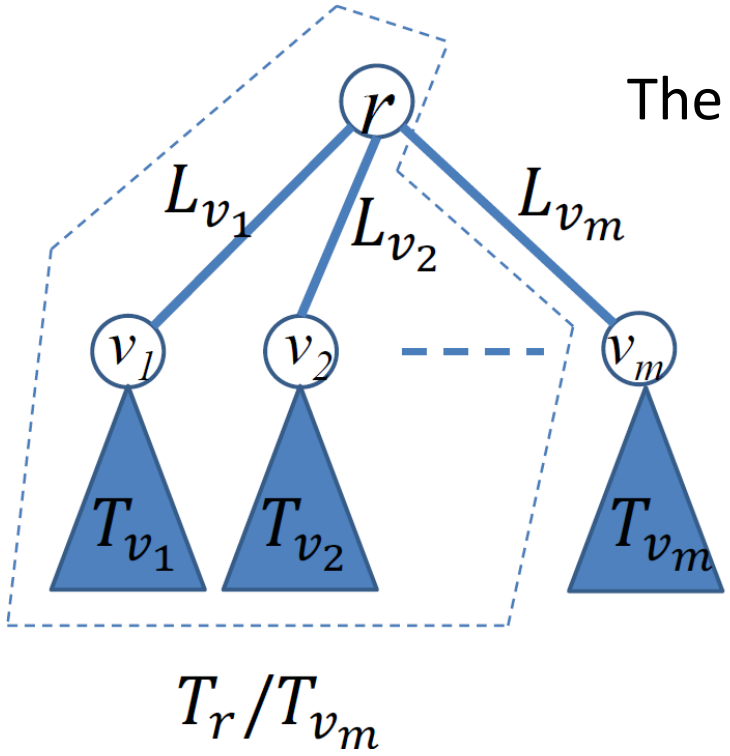
$$O_L = \frac{1}{C_L}\left(D_L + \sum_{i=1}^{K} E_i^L\right)$$



Find valid allocation while minimizing the maximum of the bandwidth occupancy ratios in the network

Allocation algorithm

- Dynamic programming
  - Minimize the maximum of the bandwidth occupancy ratios in the network



The minimum value for the allocation of N VMs to the tree T_r

$$Opt(T_r, N) = \min_x \max\{Opt(T_{vm}, x), Opt(T_r \setminus T_{vm}, N-x), O_{Lvm}(N, x)\}$$

The minimum value for the allocation of x VMs to T_{vm}

The minimum value for the allocation of N-x VMs to $T_r \setminus T_{vm}$

The bandwidth occupancy ratio of link L_{vm} for this allocation.

VM allocation for heterogeneous bandwidth demand

- First Fit (FF) algorithm
  - greedily and sequentially assigns VMs into each subtree in ascending order of their bandwidth demands

- Dynamic programming optimization for FF
  - Given a sequence of *N* VMs, decide the substring assigned to each subtree.
  - Similar to the DP allocation algorithm for homogeneous bandwidth demands.

Evaluation

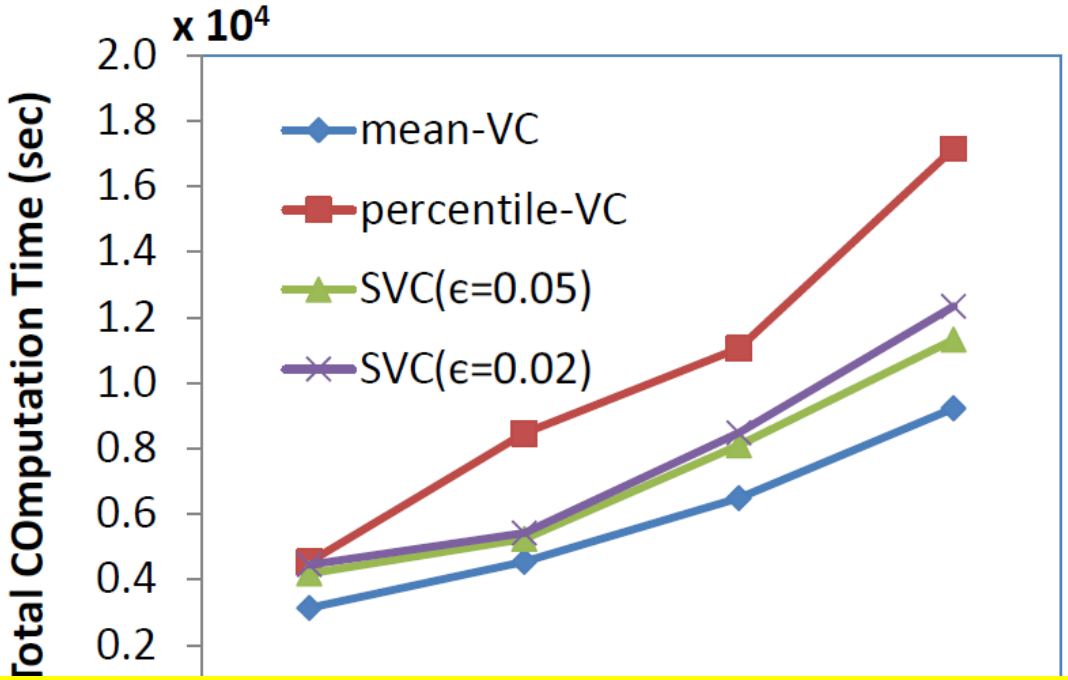
- Simulation
  - Datacenter
    - three-level tree topology
    - Rack: 20 machines, each 4 VM slots, 1Gbps link to ToR
    - 10 ToR are connected to a level-2 aggregation switch
    - 5 aggregation switches are connected to the datacenter core switch

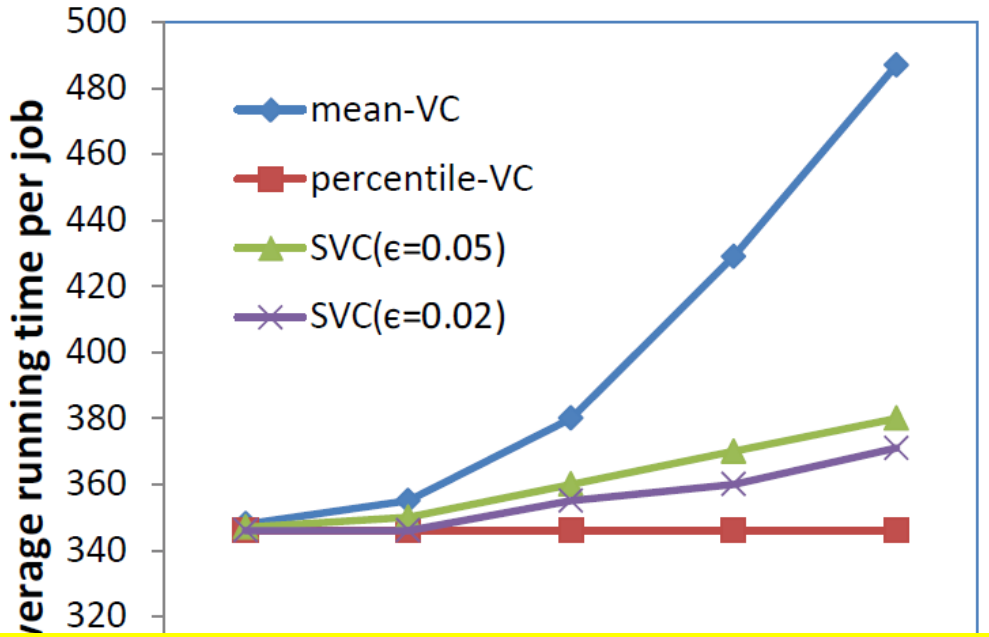
Evaluation

- Alternate abstractions: Given the normal distribution of bandwidth demand,
  - Mean-VC: the mean as the requested bandwidth in virtual cluster (VC) abstraction of Oktopus
  - Percentile-VC: 95-th percentile of the bandwidth demand as the bandwidth in VC

Evaluation

- Batched jobs: FIFO





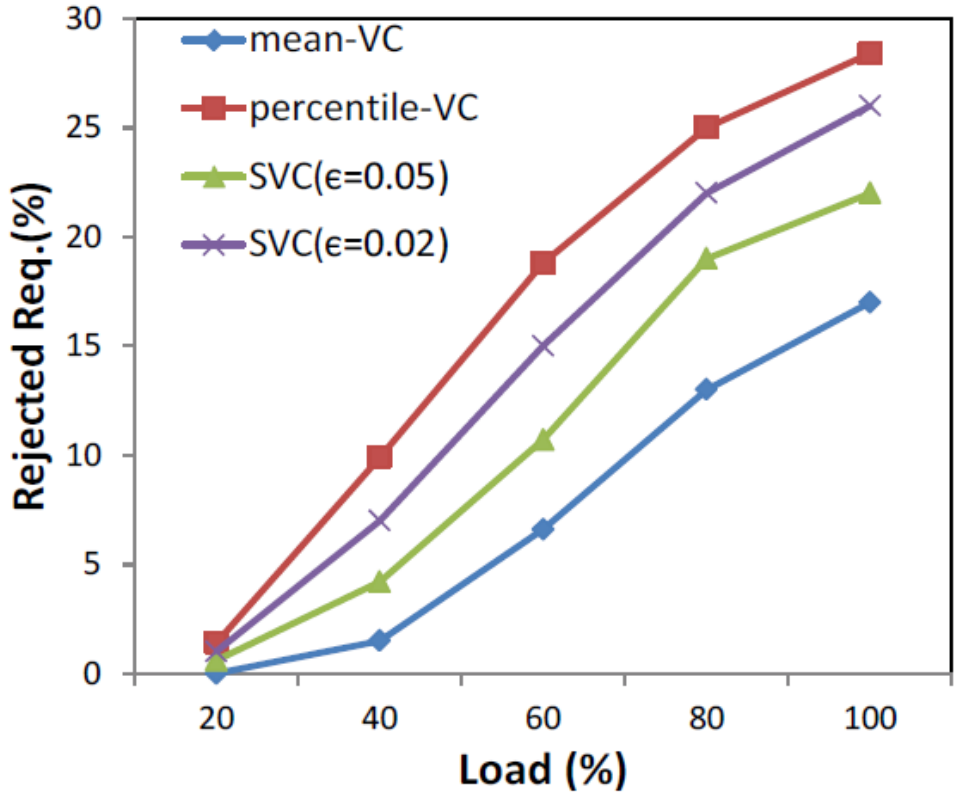
Mean-VC has lowest total completion time, but worst average running time per job;

Percentile-VC is the opposite, because it reserves a larger fixed bandwidth, which reduces flow transmission time and thus per job completion time, but decreases the job concurrency and thus increases the total batch completion time.

SVC achieves the trade-off between total completion time and average job running time.

Evaluation

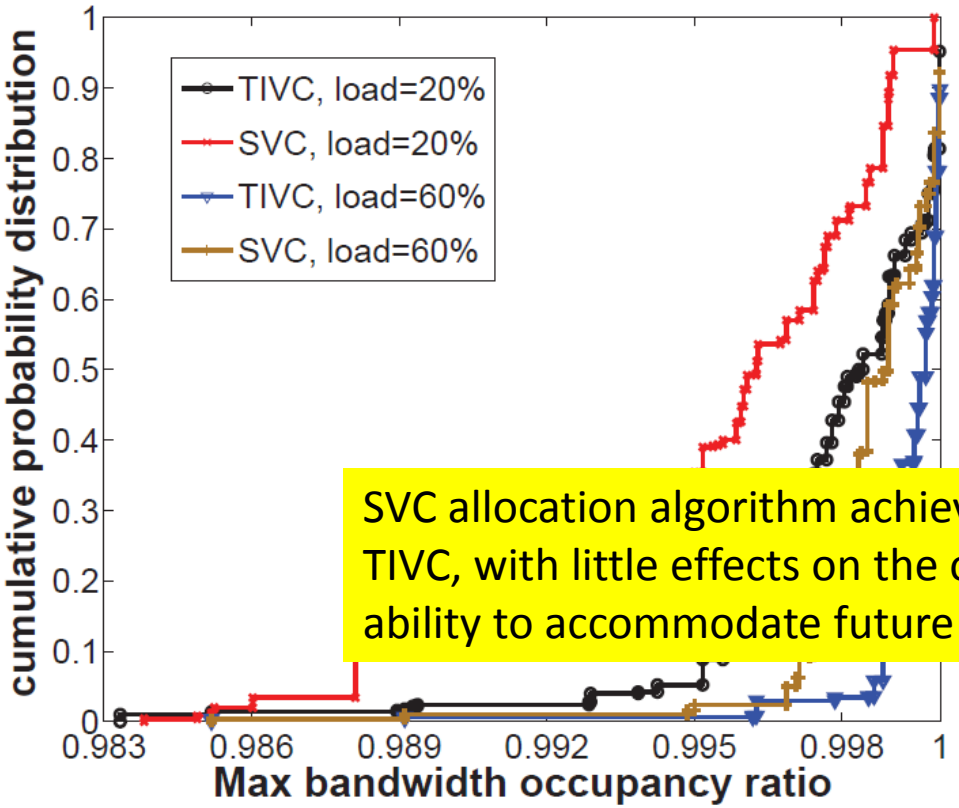
- Dynamically arriving jobs

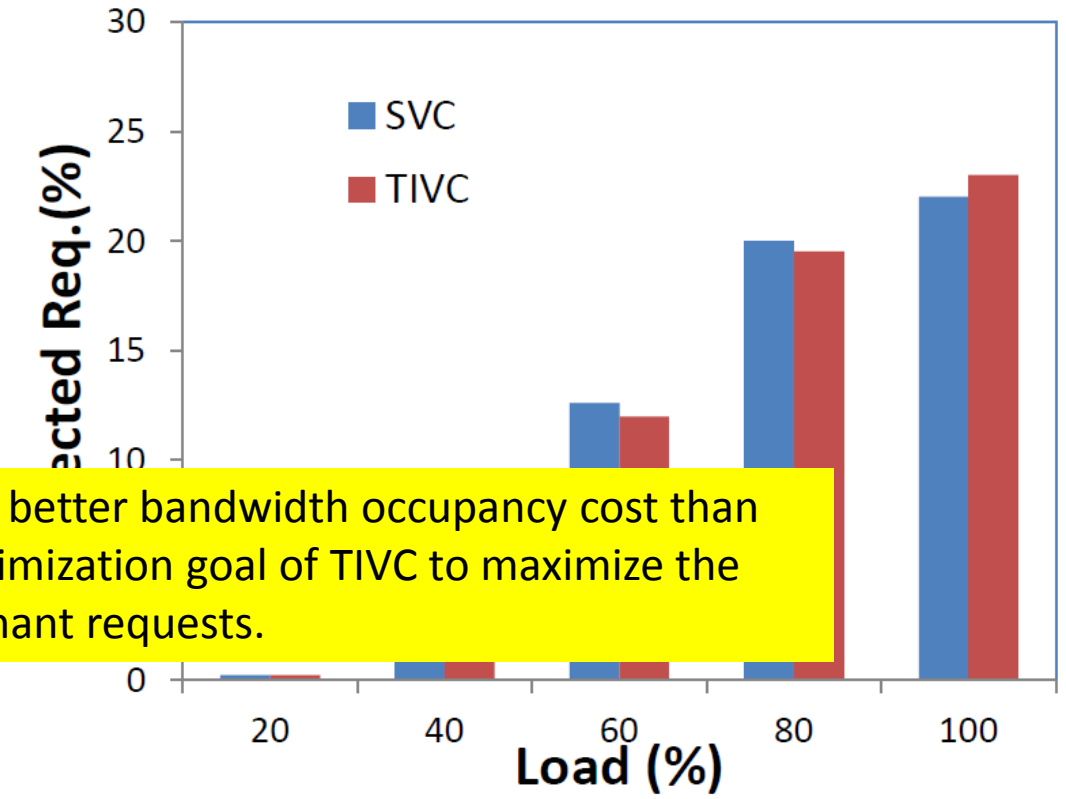


The job concurrency of SVC is between percentile-VC and mean-VC.

Evaluation

- Allocation Algorithm performance
  - Dynamically arriving jobs





SVC allocation algorithm achieves better bandwidth occupancy cost than TIVC, with little effects on the optimization goal of TIVC to maximize the ability to accommodate future tenant requests.

Conclusion and Future Work

- SVC provides probabilistic bandwidth guarantee.
- SVC achieves the trade-off between the job concurrency and the job running time for workloads with highly volatile bandwidth demands.
- Next we will characterize the probability distributions of bandwidth demands from a variety of real workloads, and implement and evaluate SVC in a real cloud environment.

Thank you!
Questions & Comments?

Haiying Shen

shenh@clemson.edu

Associate Professor of Electrical and Computer Engineering

Clemson University