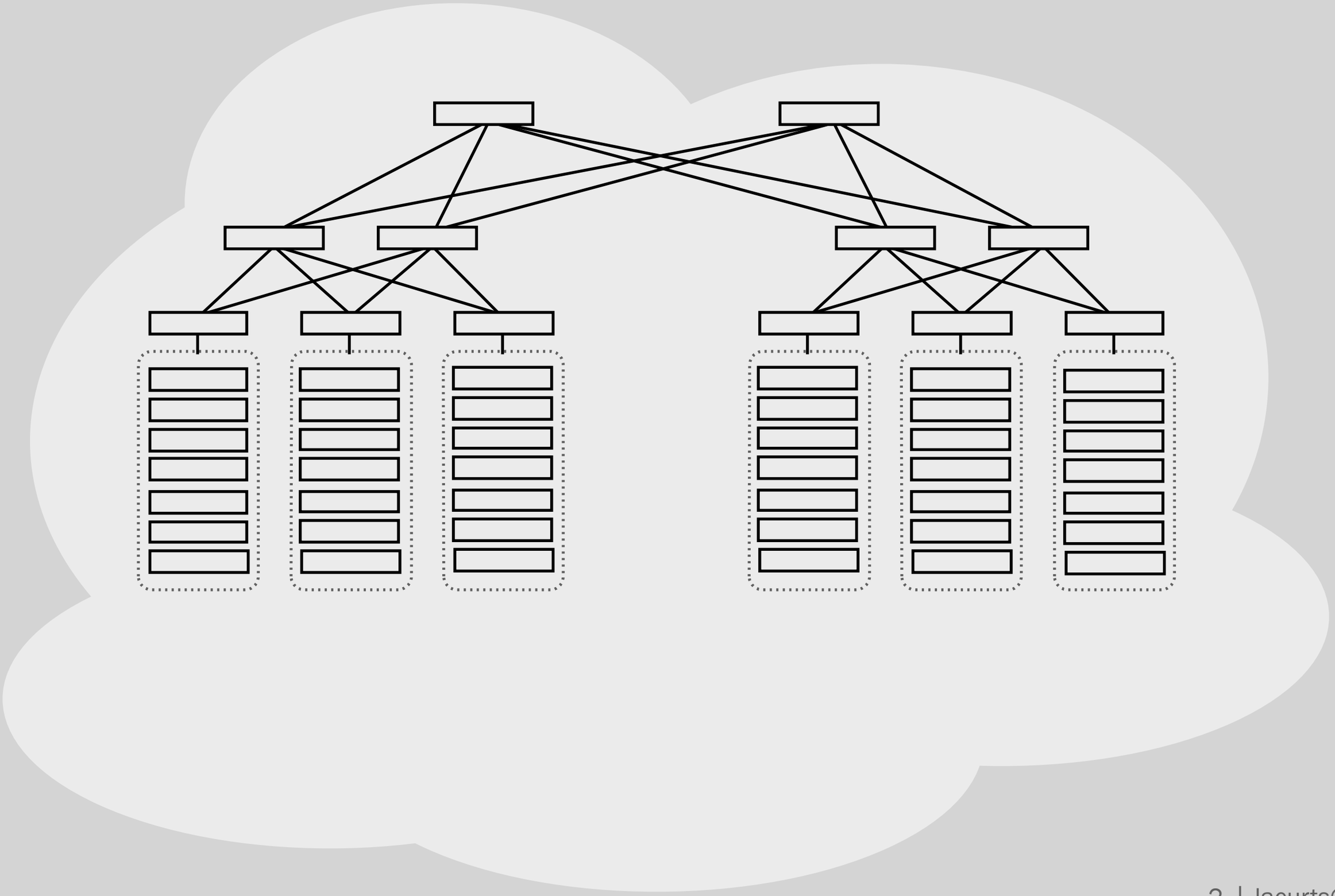
CICADA

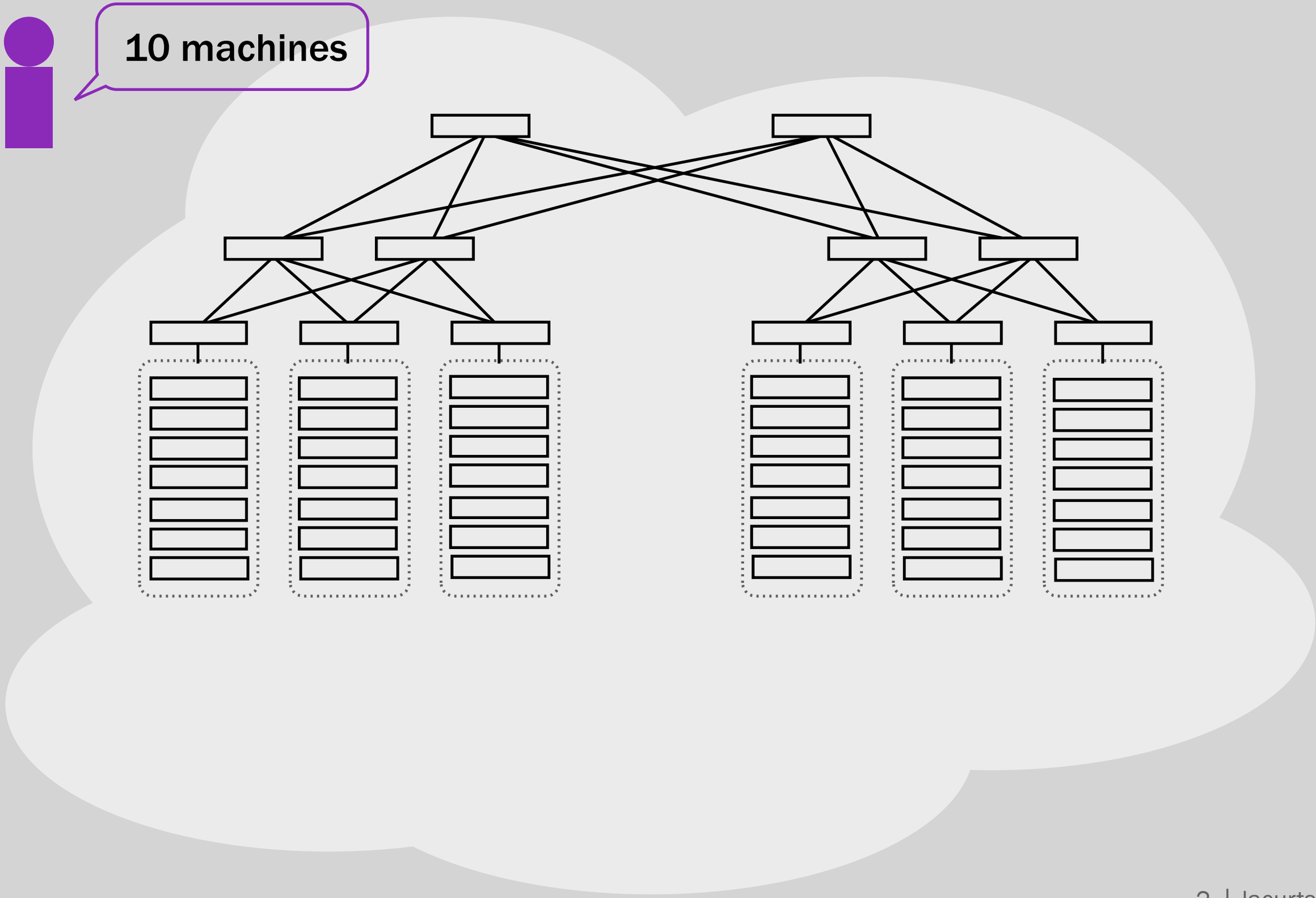
PREDICTIVE GUARANTEES FOR CLOUD NETWORKS

Katrina LaCurts
Jeff Mogul
Hari Balakrishnan
Yoshio Turner
MIT CSAIL, Google Inc., HP Labs

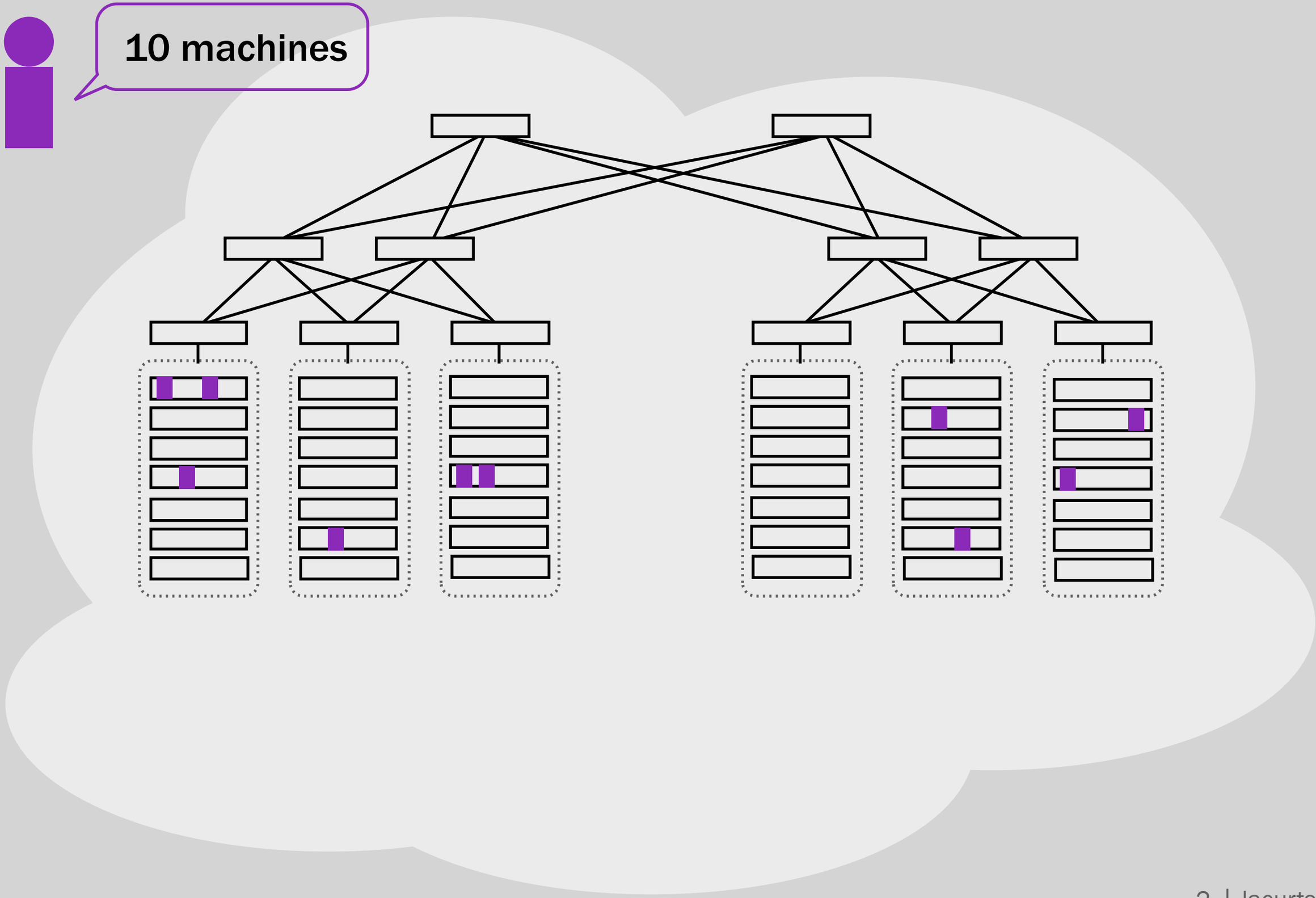
BANDWIDTH GUARANTEES



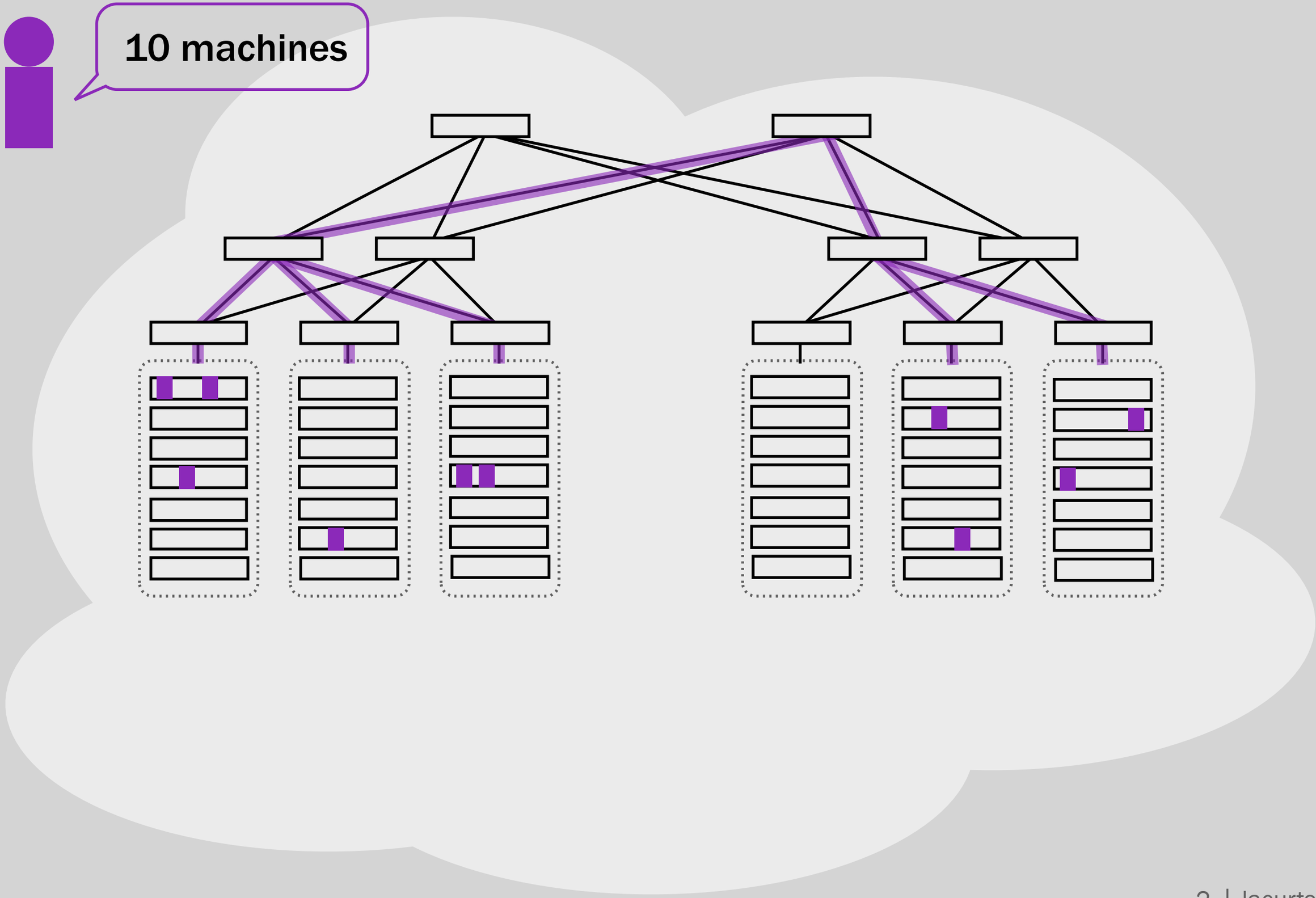
BANDWIDTH GUARANTEES



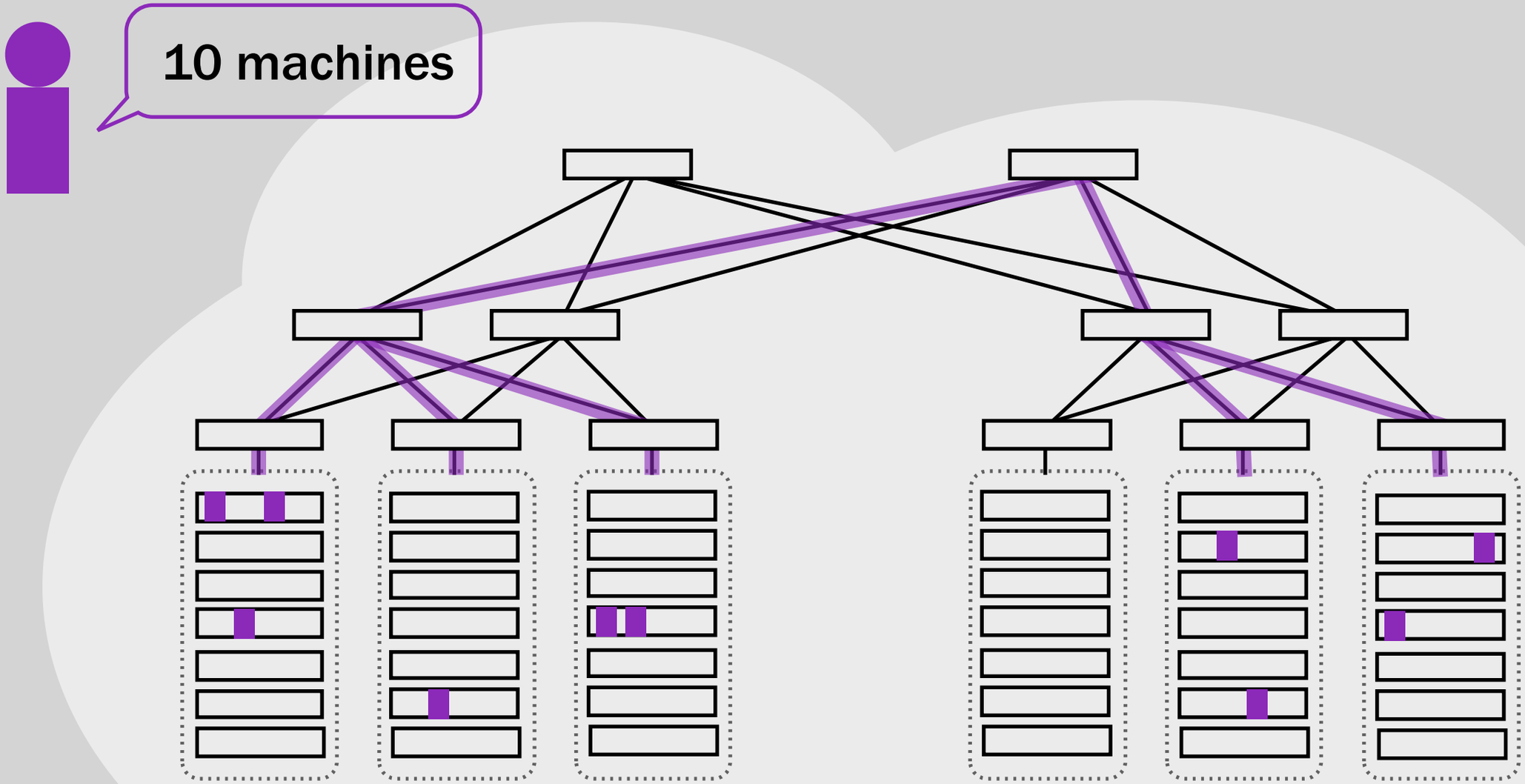
BANDWIDTH GUARANTEES



BANDWIDTH GUARANTEES



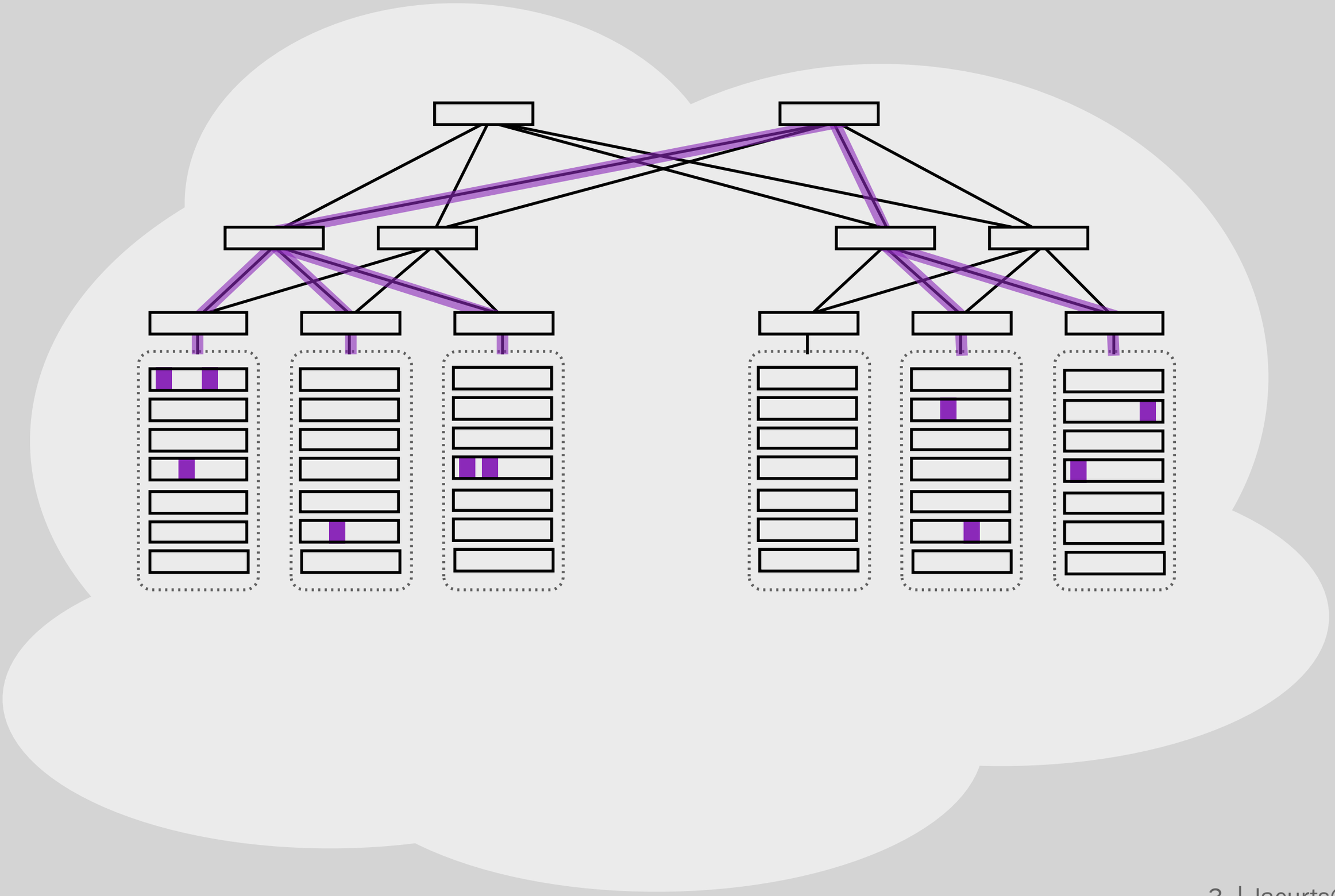
BANDWIDTH GUARANTEES



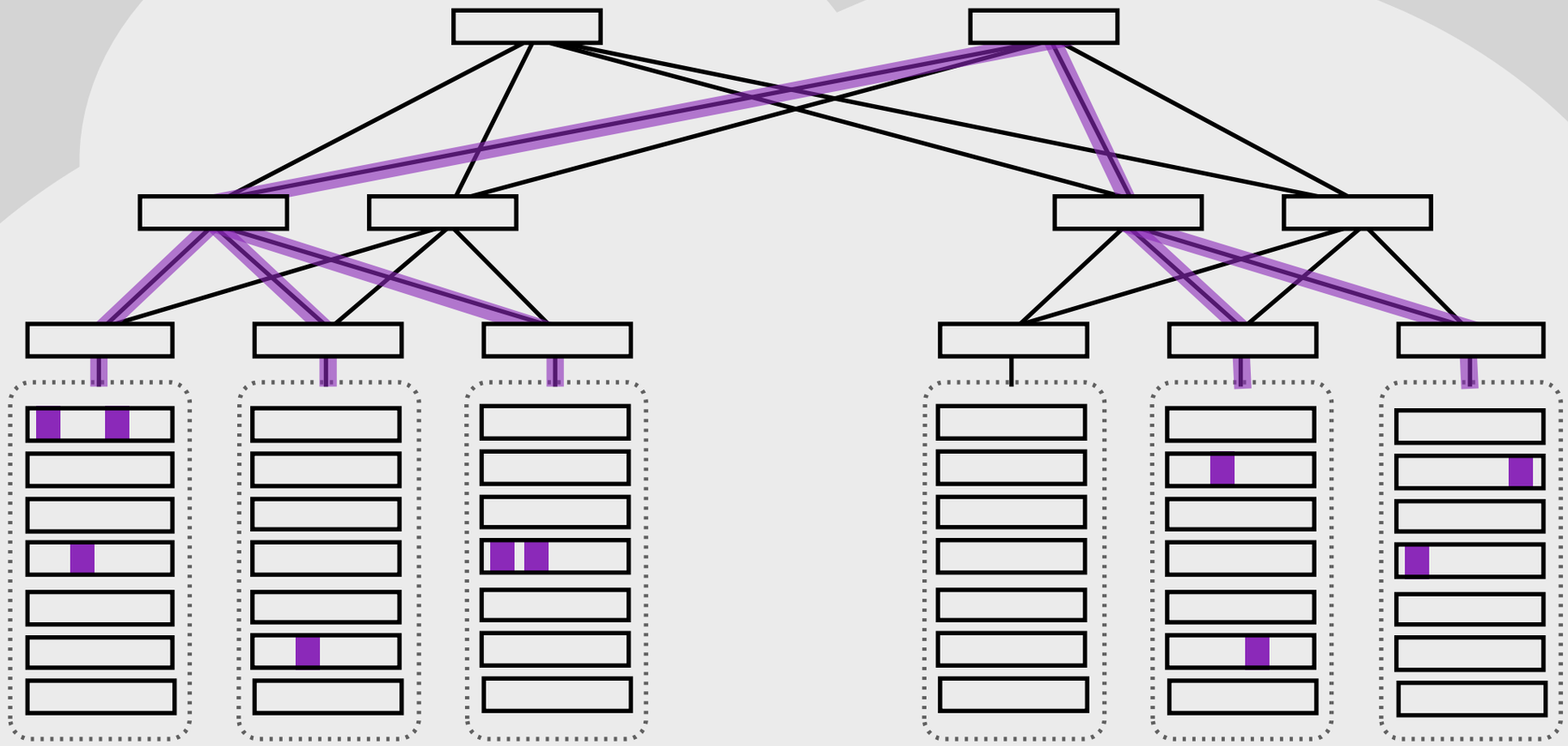
bandwidth guarantee: a promise from the provider to the customer that its VMs will be able to communicate with each other at a particular rate

(informal definition)

WHY GUARANTEES?



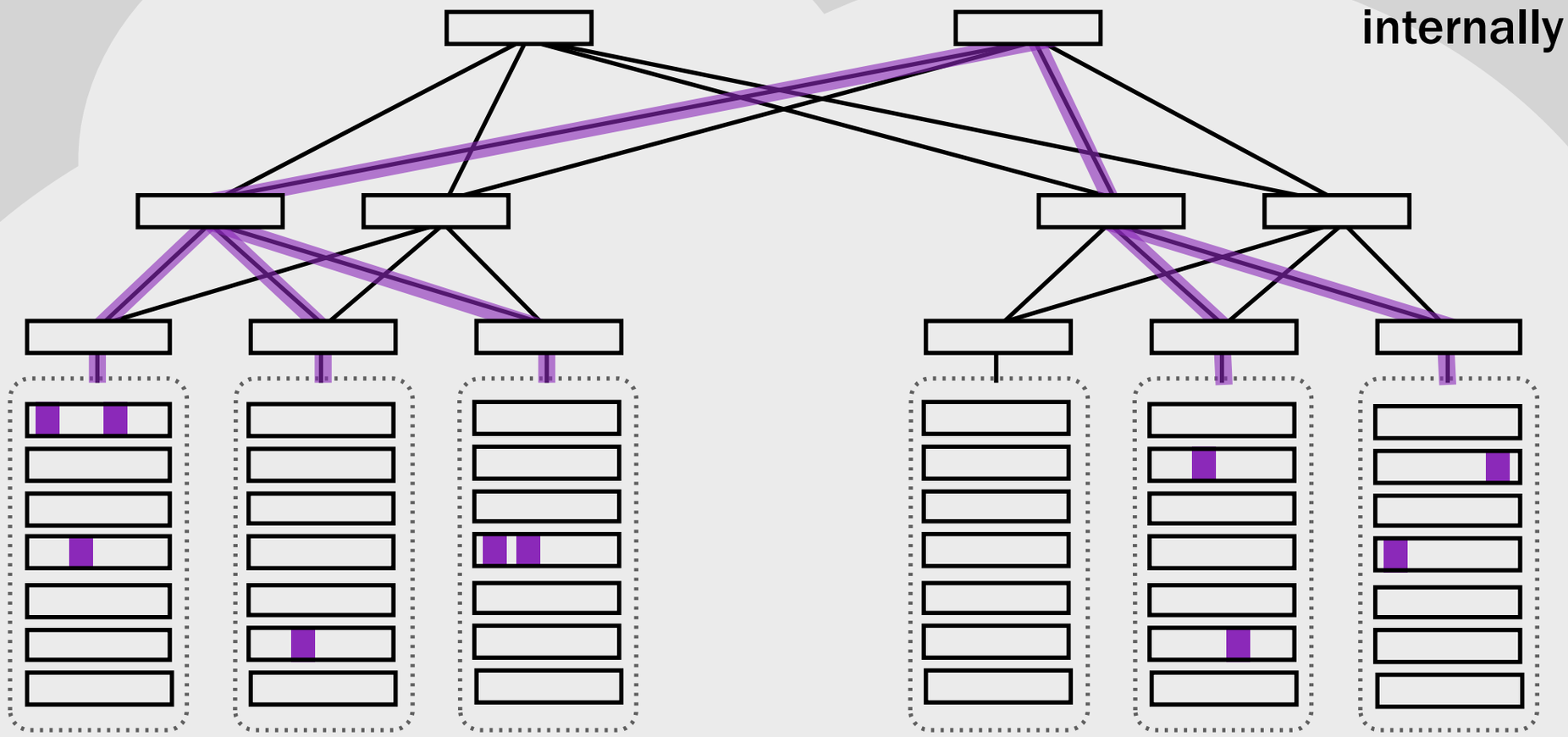
WHY GUARANTEES?



is there really that much data?

WHY GUARANTEES?

applications can send terabytes or petabytes of data internally [1]

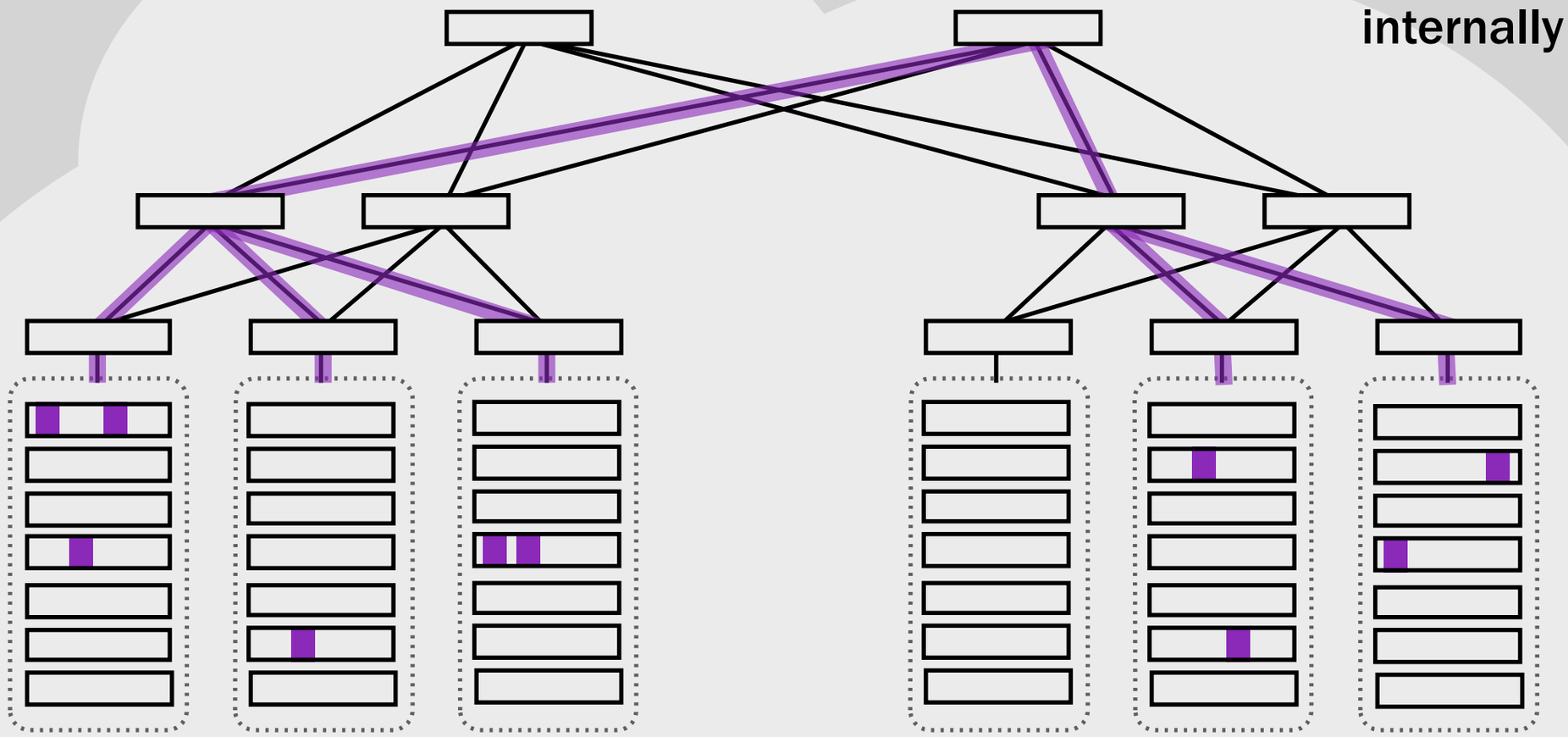


is there really that much data?

[1] Zaharia, et al. “Improving MapReduce Performance in Heterogeneous Environments”. OSDI 2008

WHY GUARANTEES?

applications can send terabytes or petabytes of data internally [1]



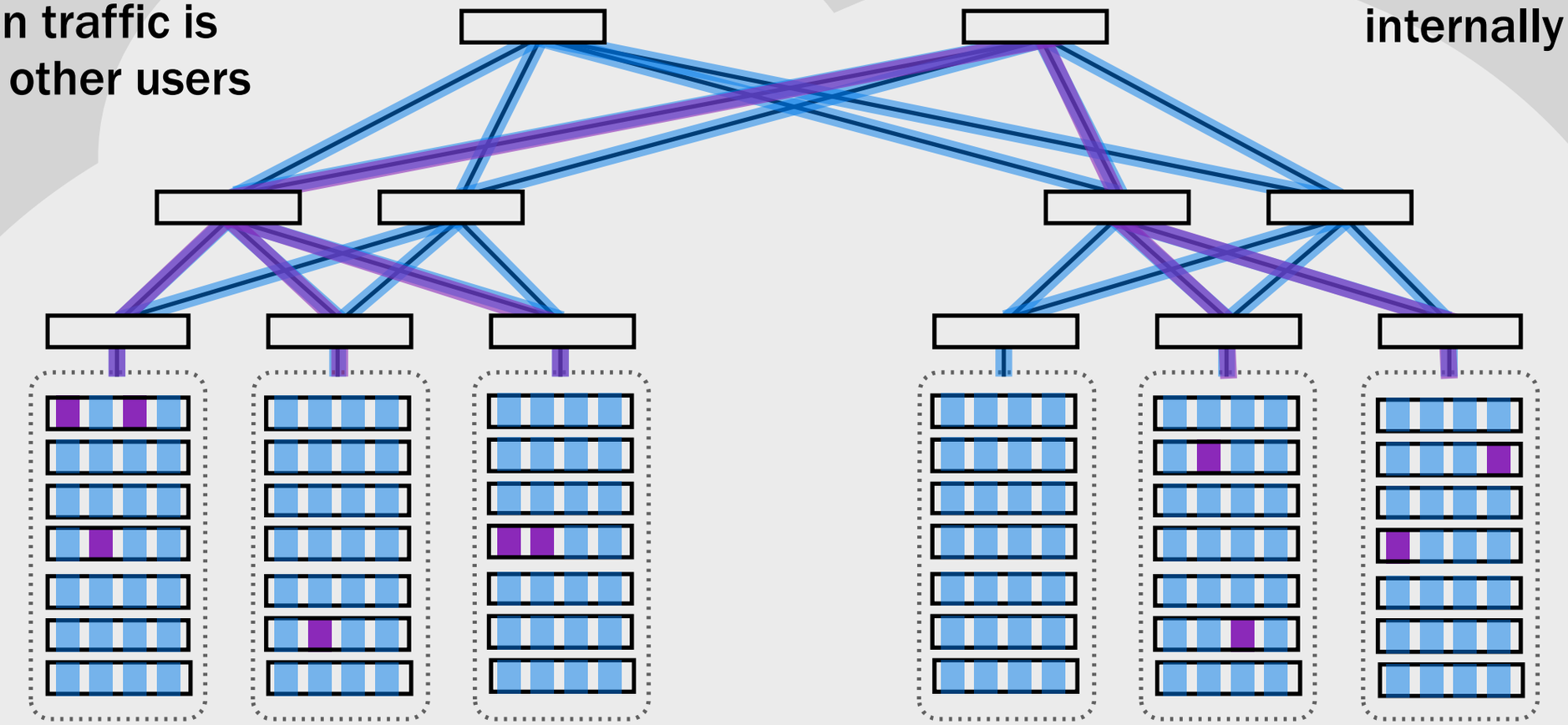
aren't cloud networks homogeneous and well-provisioned?

[1] Zaharia, et al. "Improving MapReduce Performance in Heterogeneous Environments". OSDI 2008

WHY GUARANTEES?

application traffic is affected by other users

applications can send terabytes or petabytes of data internally [1]



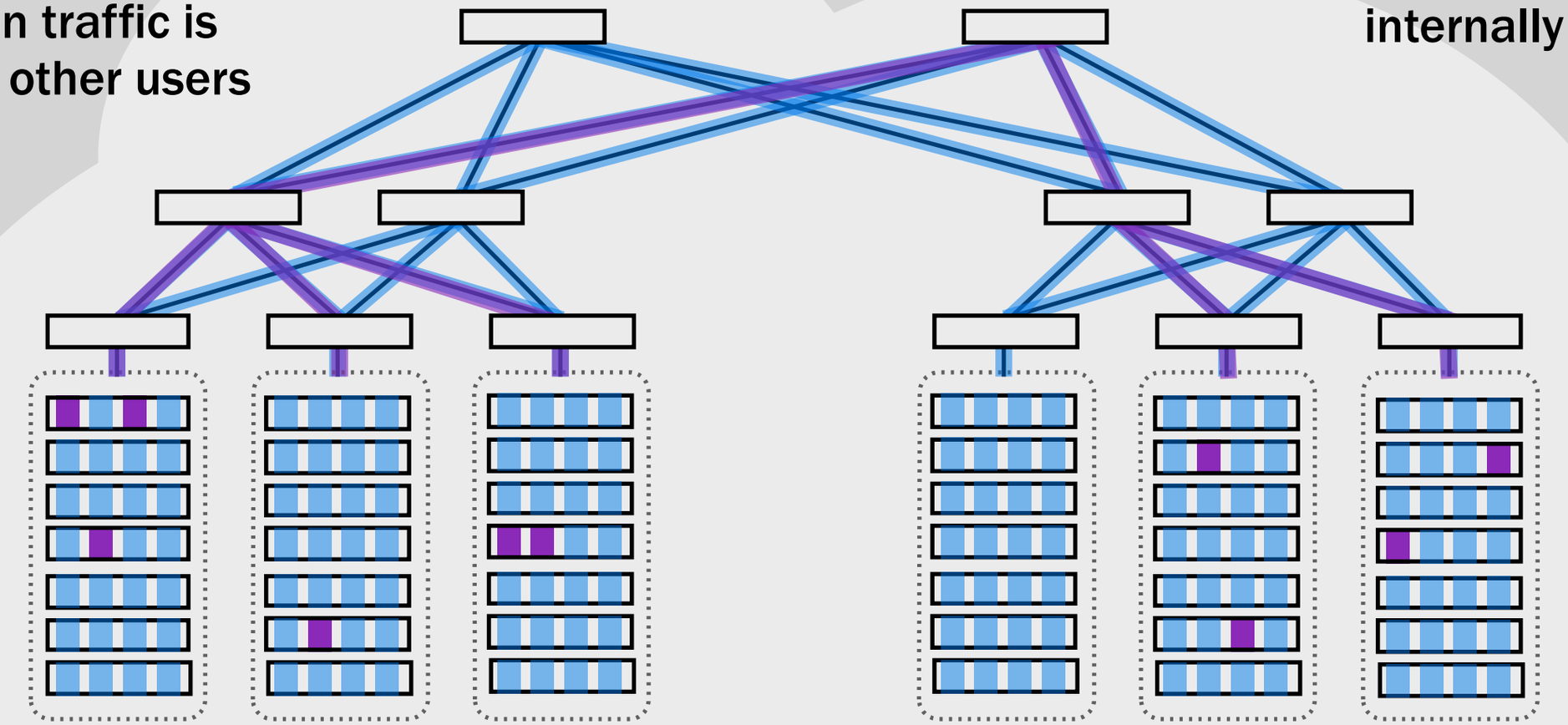
aren't cloud networks homogeneous and well-provisioned?

[1] Zaharia, et al. "Improving MapReduce Performance in Heterogeneous Environments". OSDI 2008

WHY GUARANTEES?

application traffic is affected by other users

applications can send terabytes or petabytes of data internally [1]



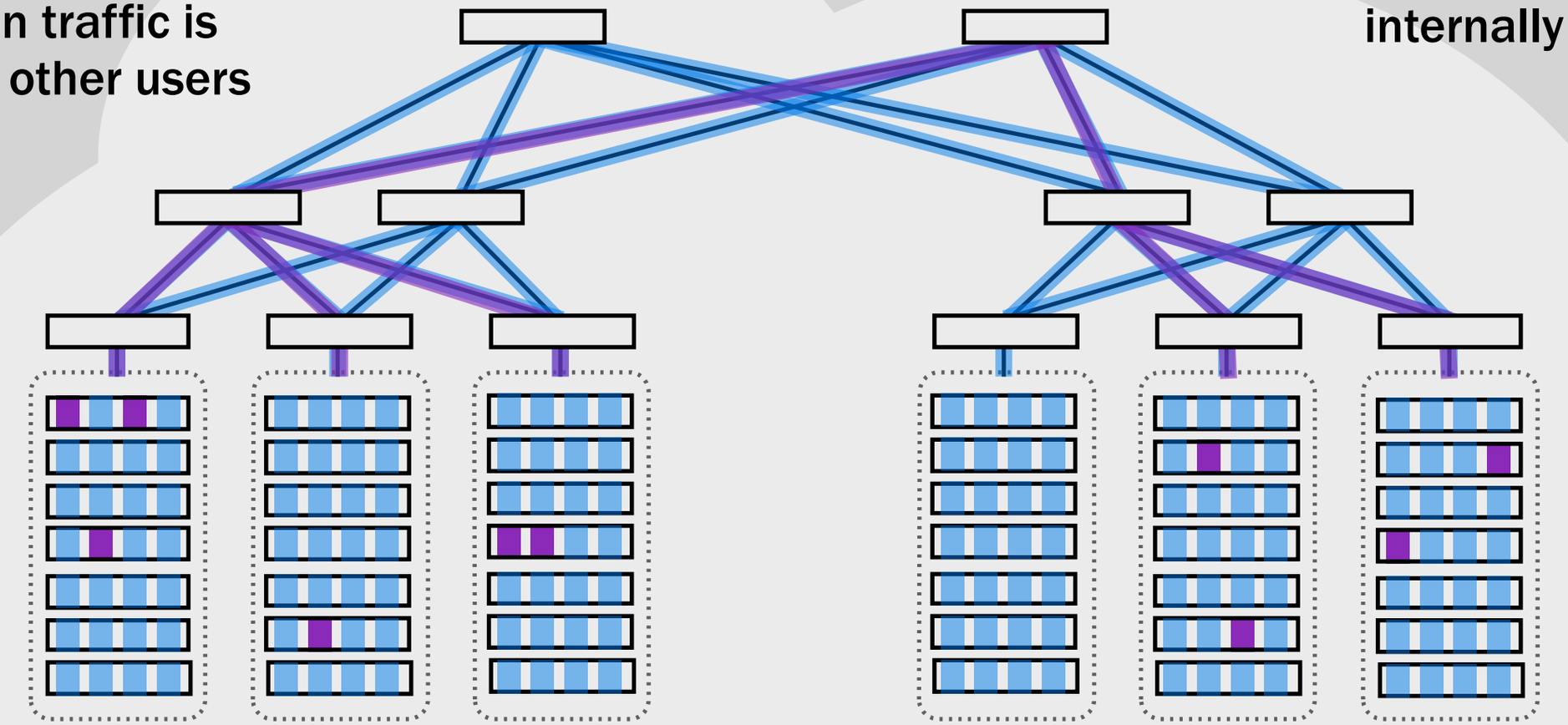
what customers care about network performance?

[1] Zaharia, et al. “Improving MapReduce Performance in Heterogeneous Environments”. OSDI 2008

WHY GUARANTEES?

application traffic is affected by other users

applications can send terabytes or petabytes of data internally [1]

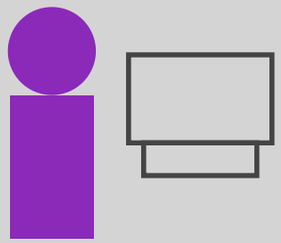


what customers care about network performance?

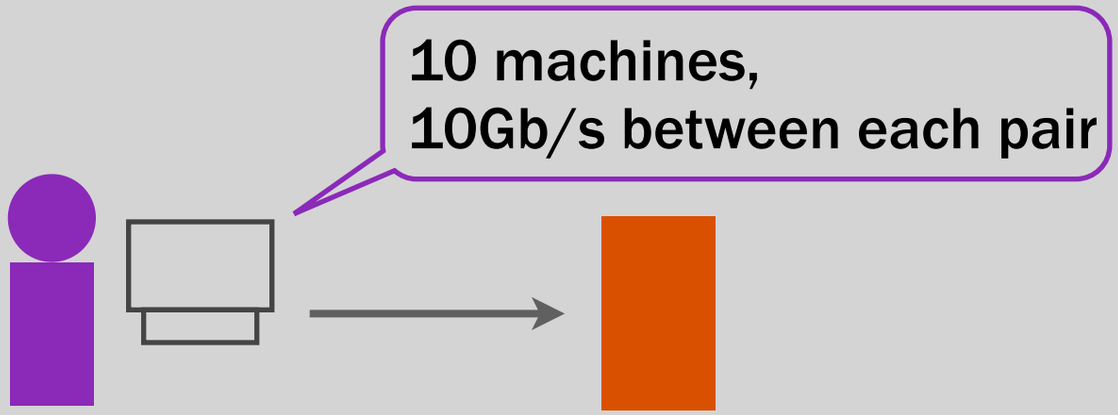
enterprise customers need to satisfy SLAs with their own customers

[1] Zaharia, et al. “Improving MapReduce Performance in Heterogeneous Environments”. OSDI 2008

POSSIBLE ARCHITECTURE

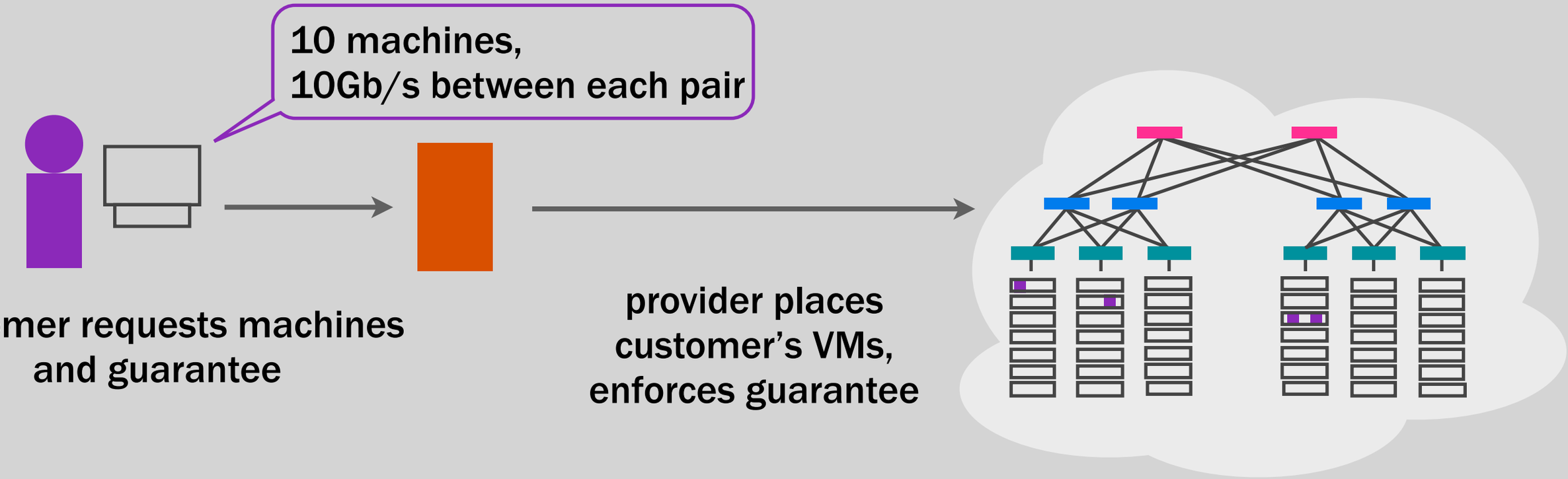


POSSIBLE ARCHITECTURE

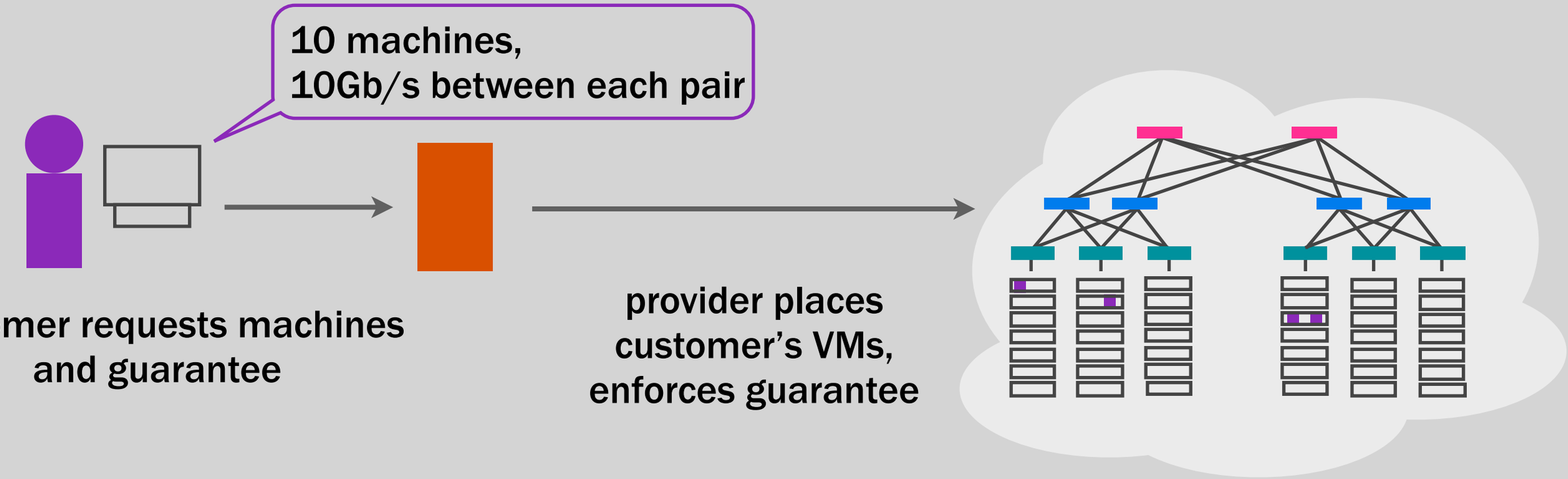


customer requests machines
and guarantee

POSSIBLE ARCHITECTURE

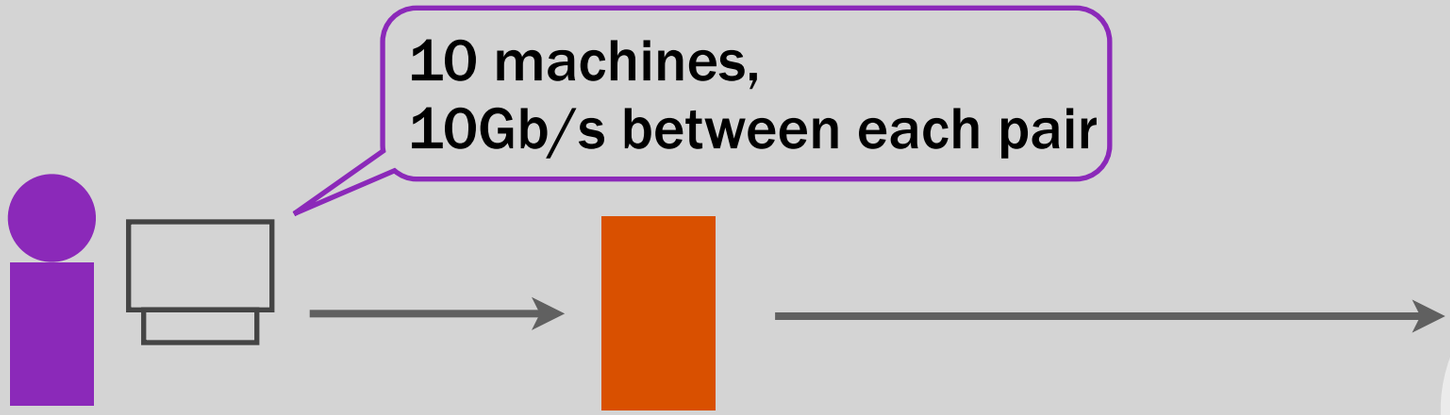


POSSIBLE ARCHITECTURE



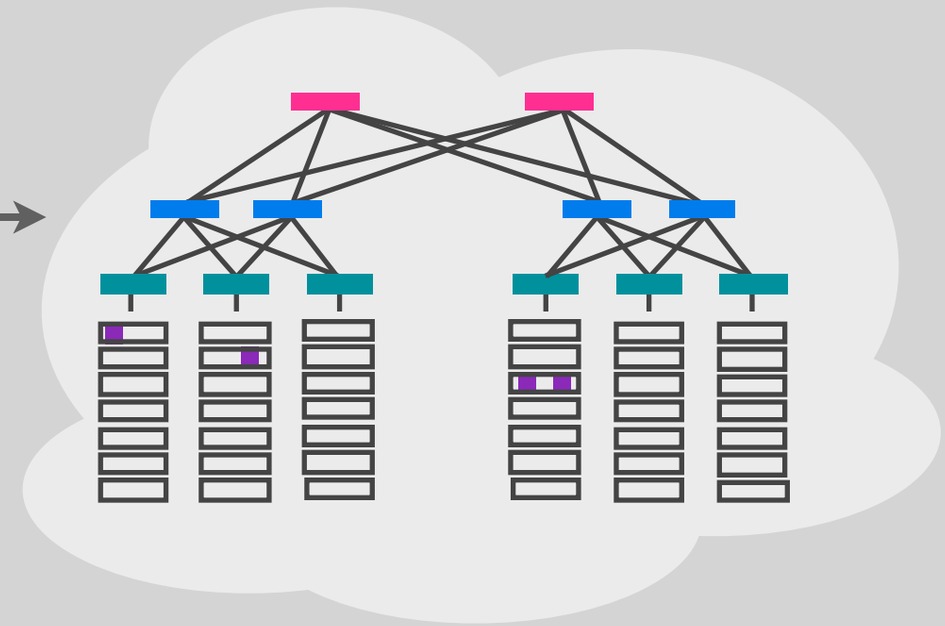
what if some pairs of VMs use
fewer than 10Gb/s?

POSSIBLE ARCHITECTURE



customer requests machines and guarantee

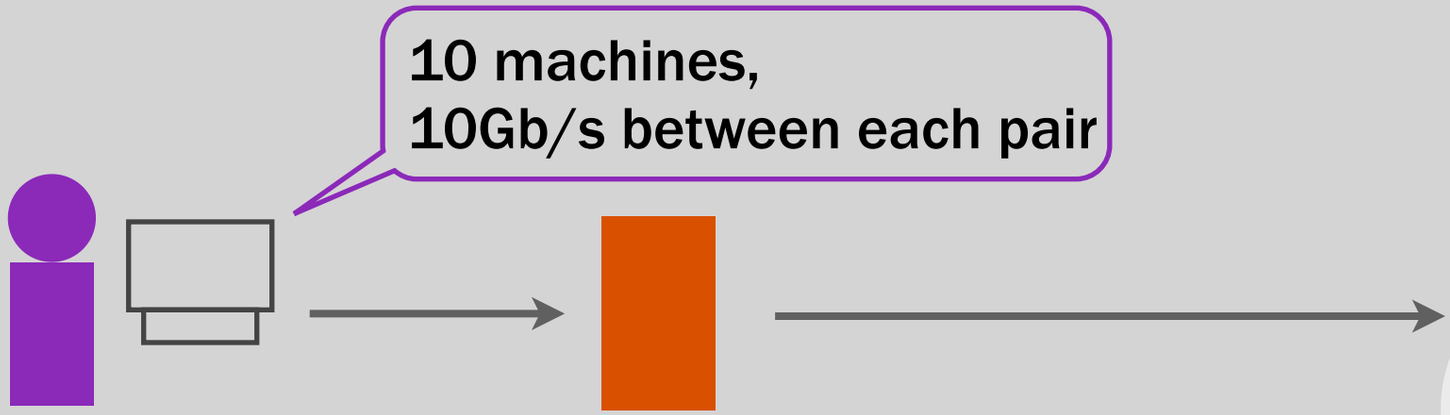
provider places customer's VMs, enforces guarantee



what if some pairs of VMs use fewer than 10Gb/s?

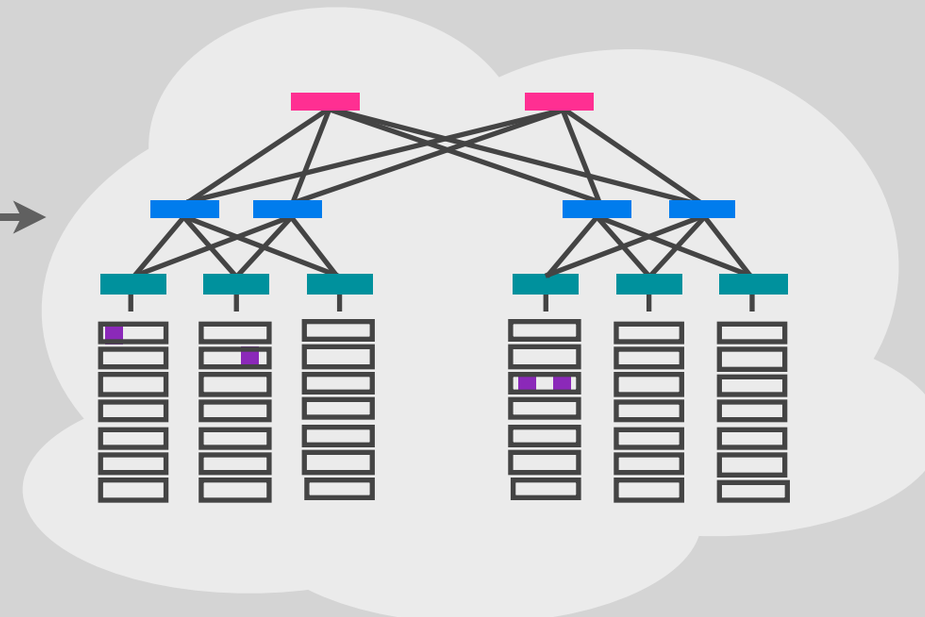
over-provisioned network
increased cost to customer,
wasted bandwidth within network

POSSIBLE ARCHITECTURE



customer requests machines and guarantee

provider places customer's VMs, enforces guarantee

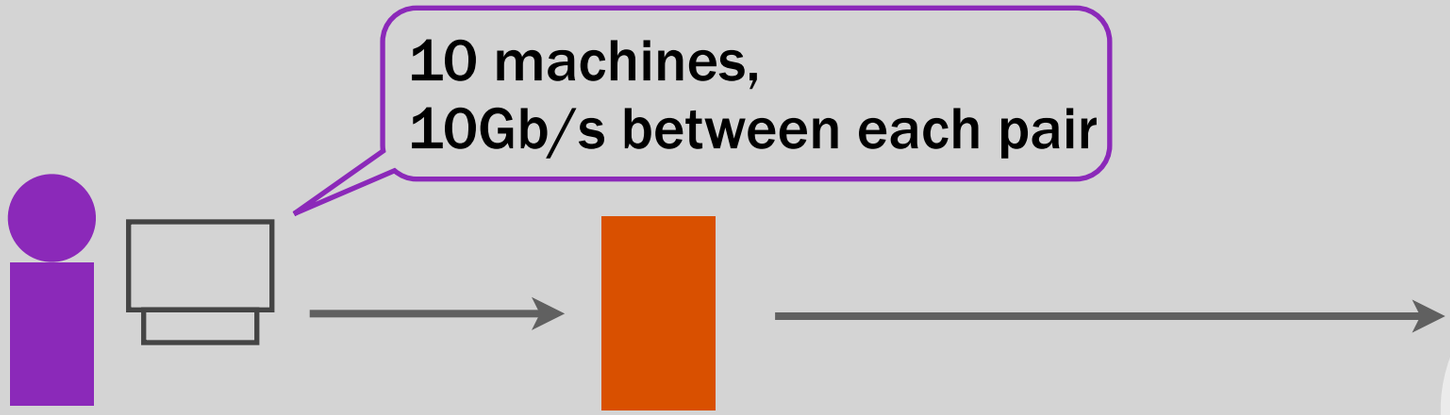


what if some pairs of VMs use fewer than 10Gb/s?

over-provisioned network
increased cost to customer, wasted bandwidth within network

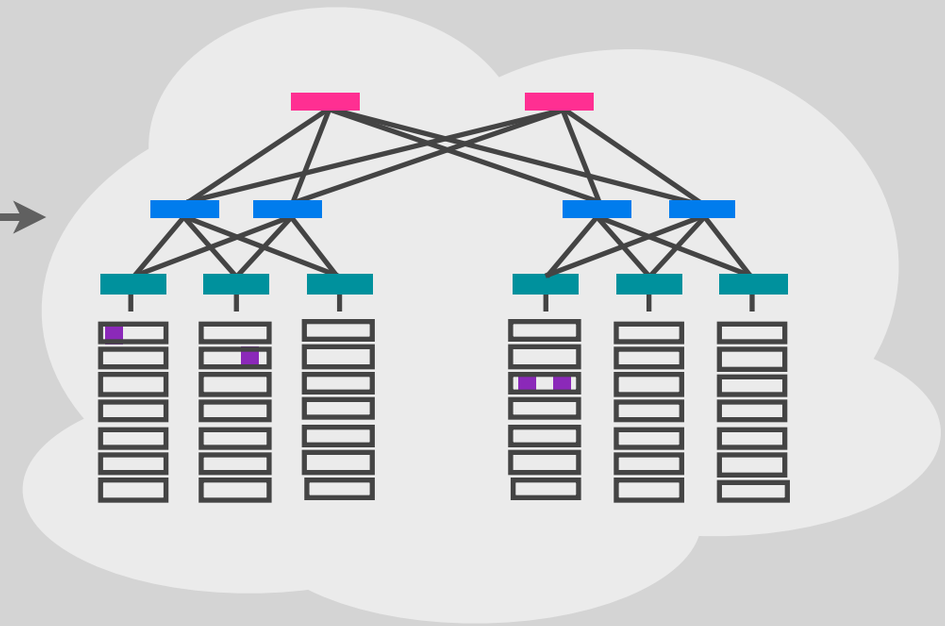
what if some pairs of VMs need more than 10Gb/s?

POSSIBLE ARCHITECTURE



customer requests machines and guarantee

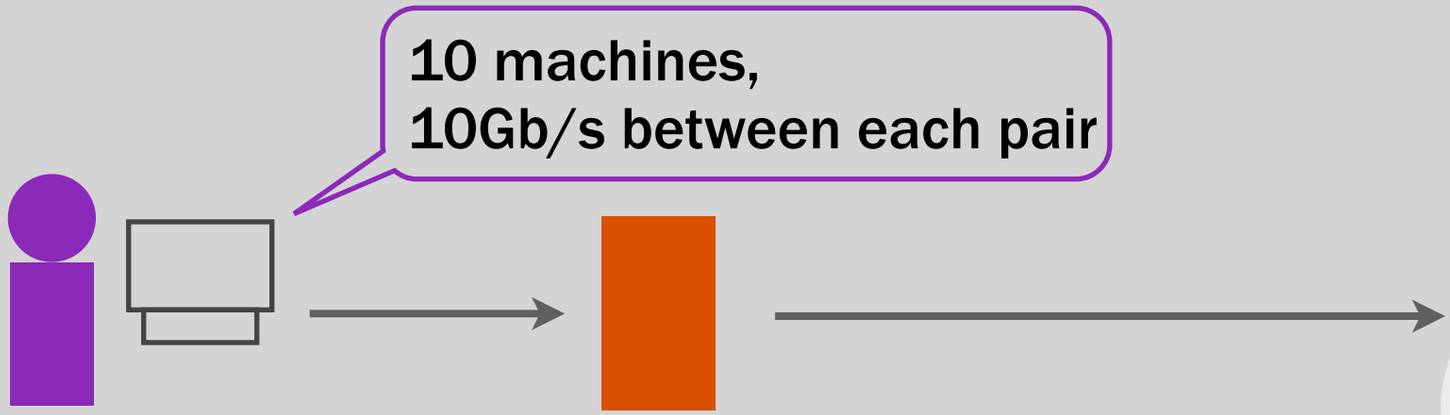
provider places customer's VMs, enforces guarantee



what if some pairs of VMs use fewer than 10Gb/s? → **over-provisioned network**
increased cost to customer, wasted bandwidth within network

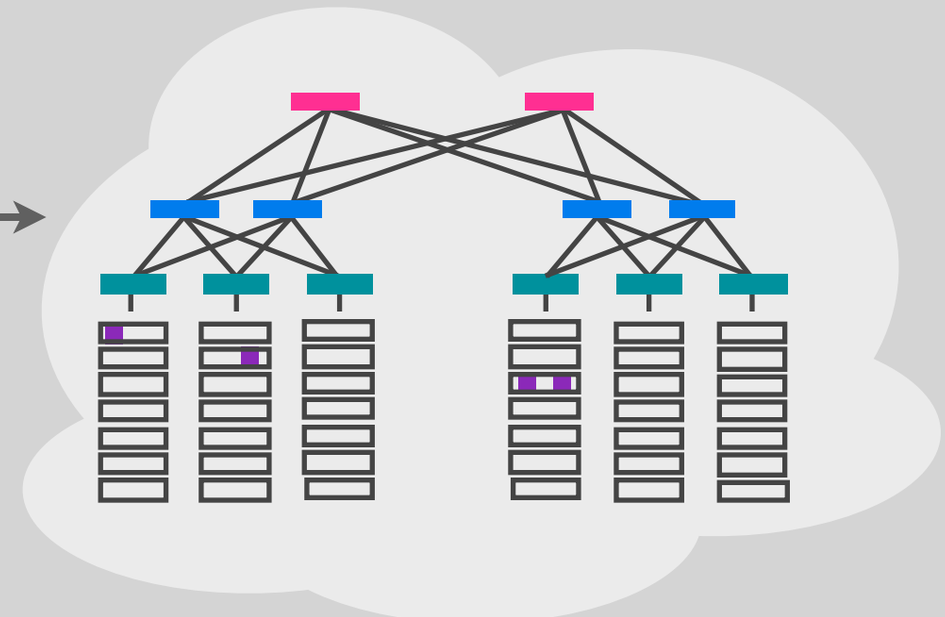
what if some pairs of VMs need more than 10Gb/s? → **under-provisioned network**
poor performance for customer's application

POSSIBLE ARCHITECTURE



customer requests machines and guarantee

provider places customer's VMs, enforces guarantee



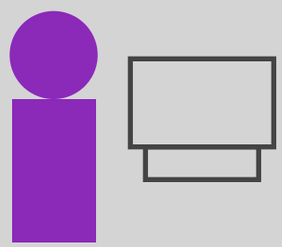
what if some pairs of VMs use fewer than 10Gb/s? → **over-provisioned network**: increased cost to customer, wasted bandwidth within network

what if some pairs of VMs need more than 10Gb/s? → **under-provisioned network**: poor performance for customer's application

problem: how do customers know what guarantee they need?

CICADA'S ARCHITECTURE

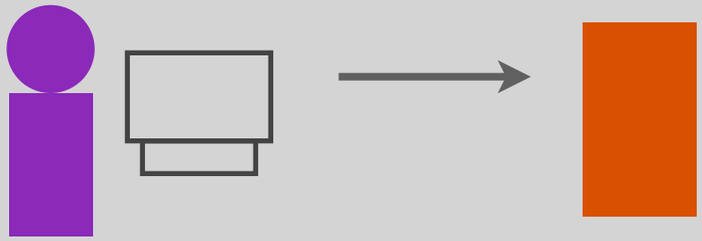
cicada makes predictions about an application's traffic to automatically generate a guarantee



CICADA'S ARCHITECTURE

cicada makes predictions about an application's traffic to automatically generate a guarantee

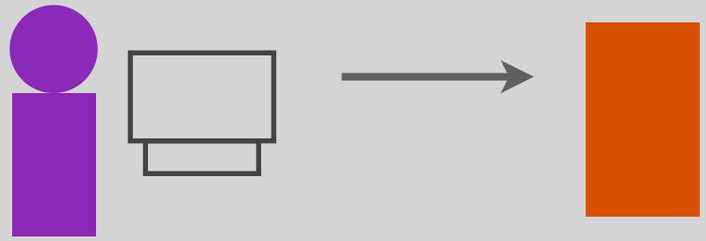
customer makes initial request for machines



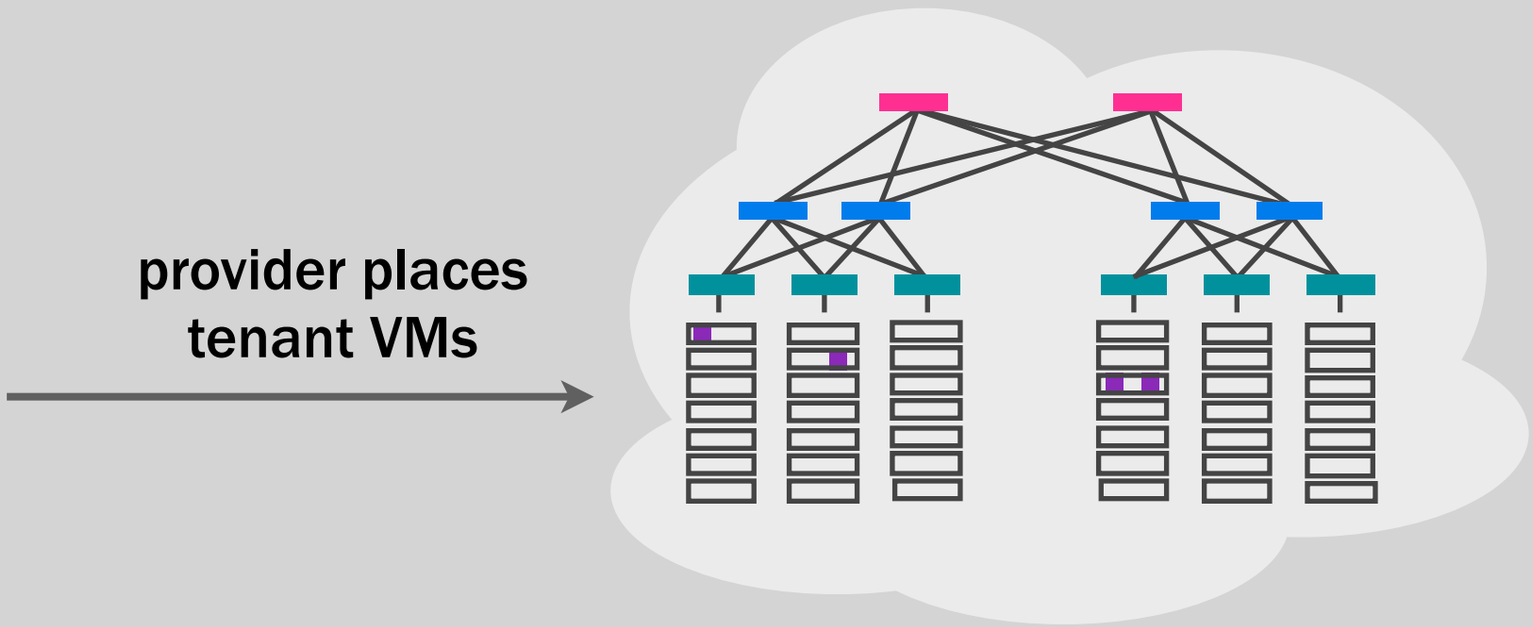
CICADA'S ARCHITECTURE

cicada makes predictions about an application's traffic to automatically generate a guarantee

customer makes initial request for machines

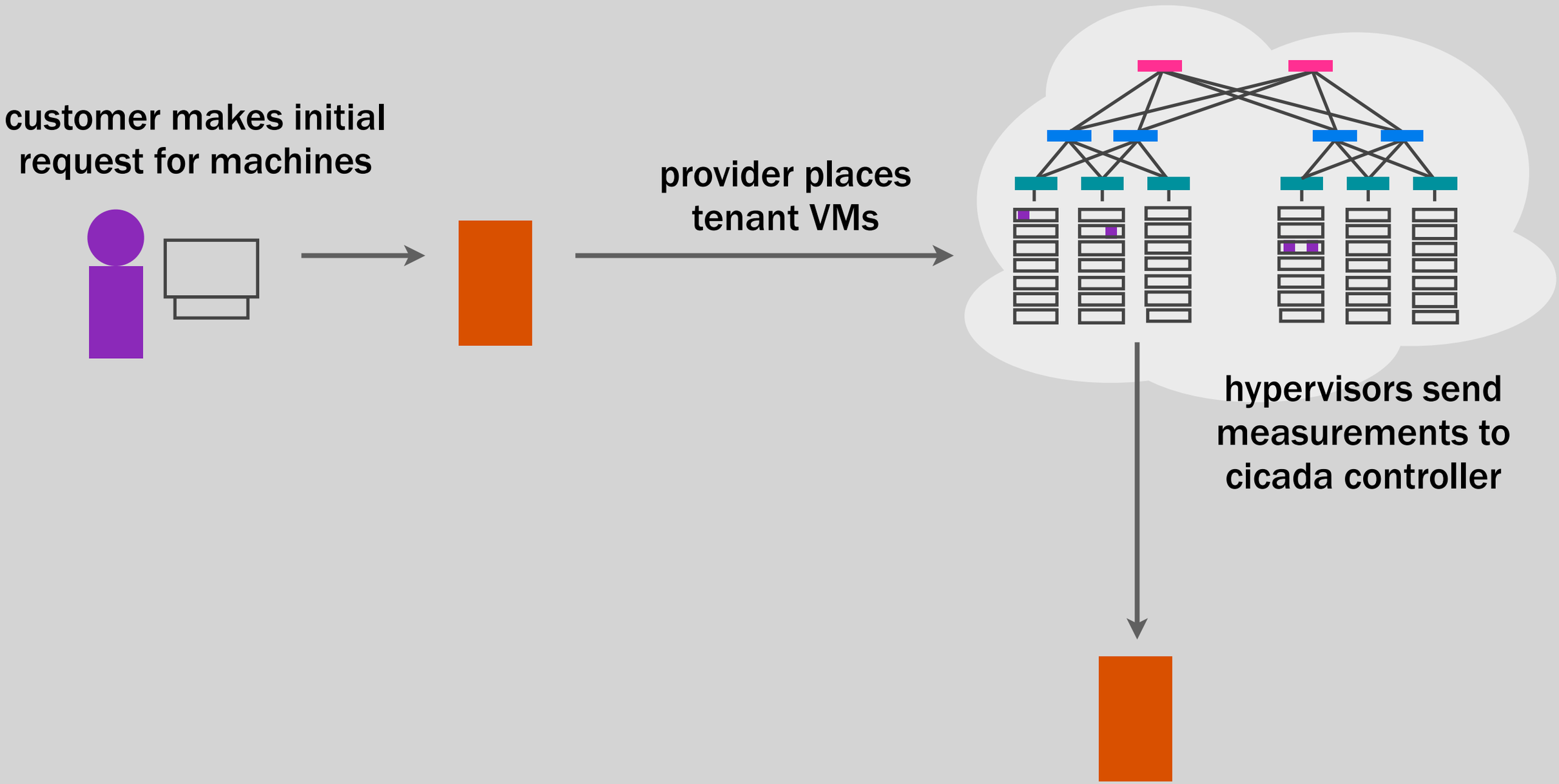


provider places tenant VMs



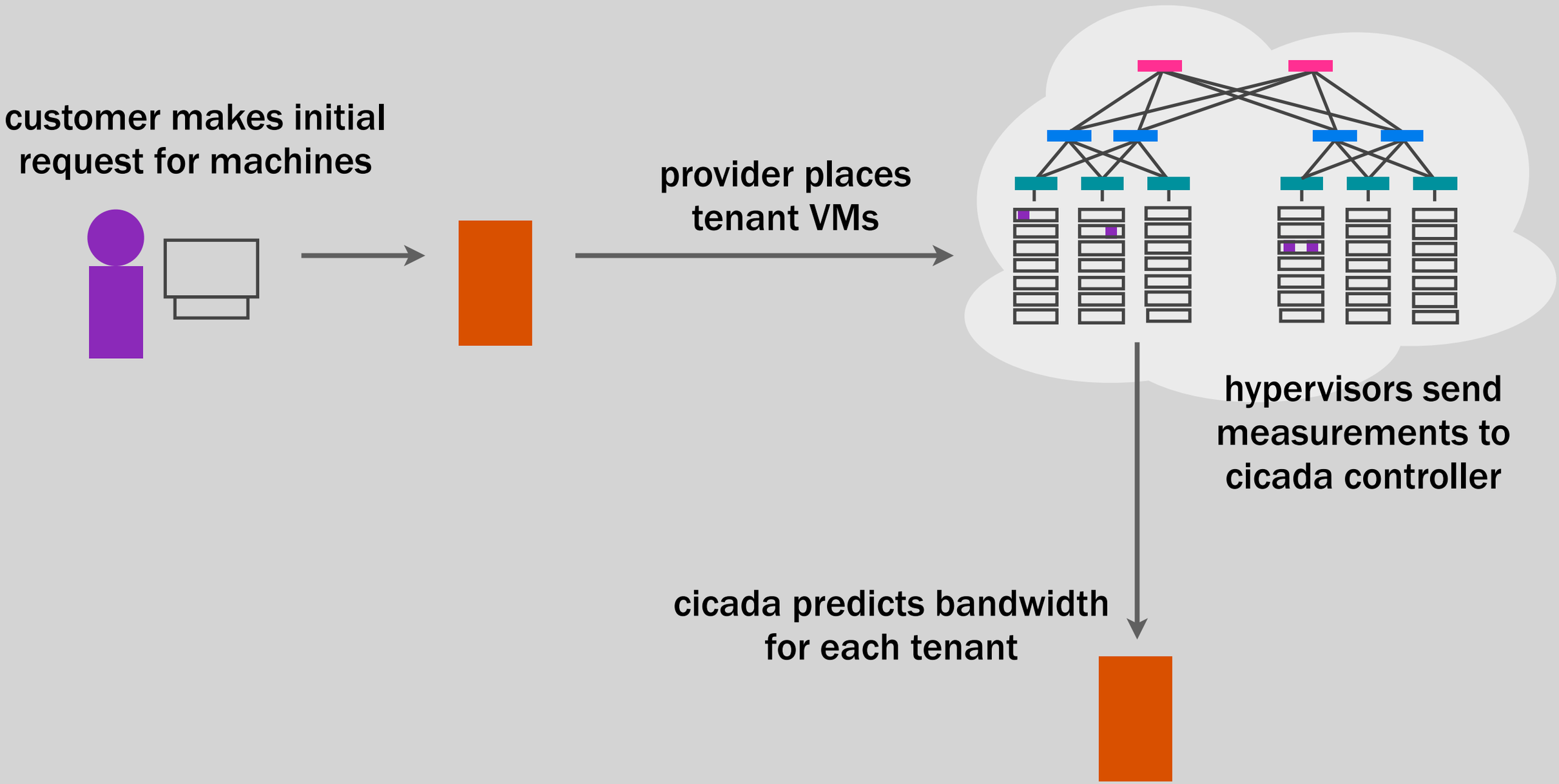
CICADA'S ARCHITECTURE

cicada makes predictions about an application's traffic to automatically generate a guarantee



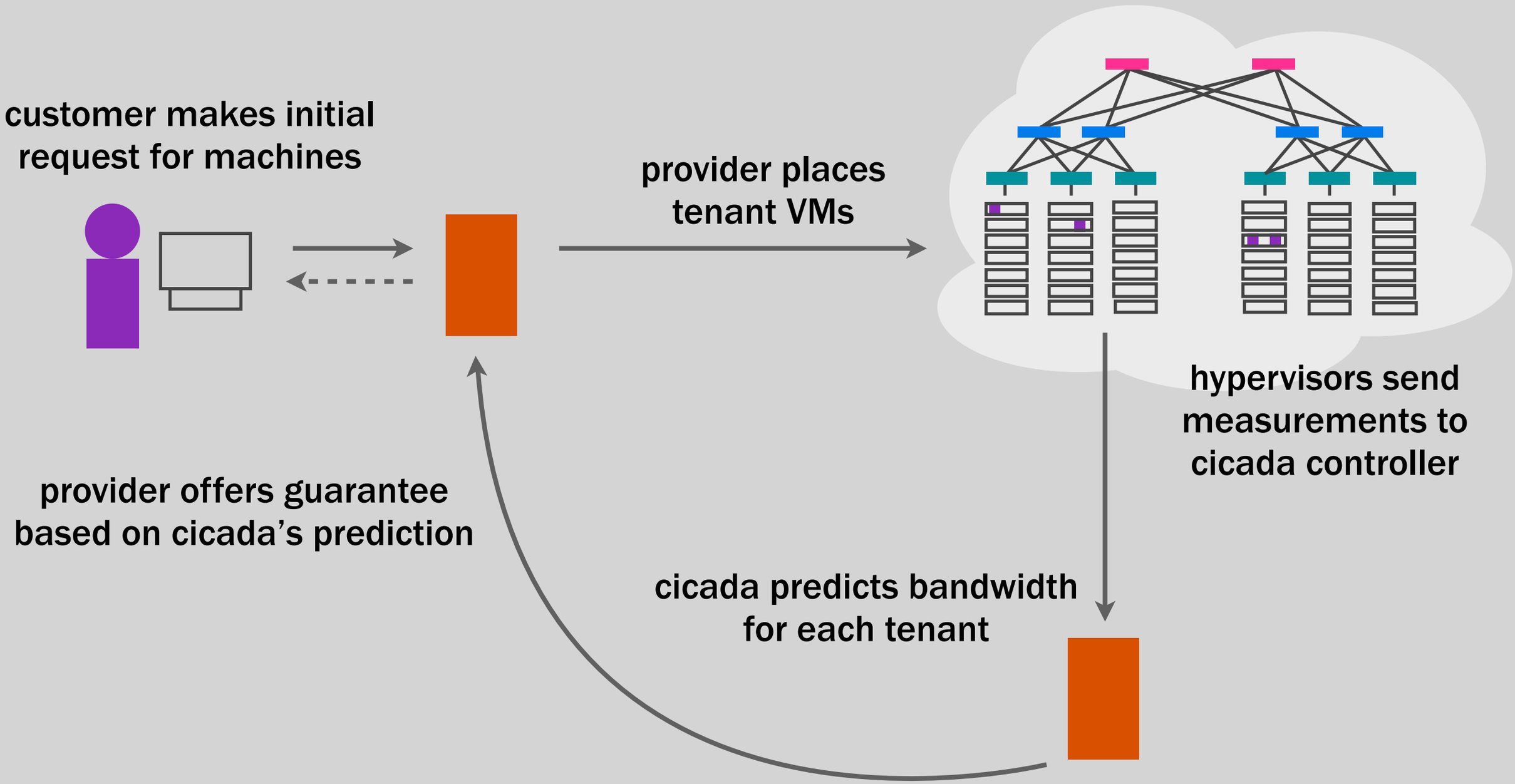
CICADA'S ARCHITECTURE

cicada makes predictions about an application's traffic to automatically generate a guarantee



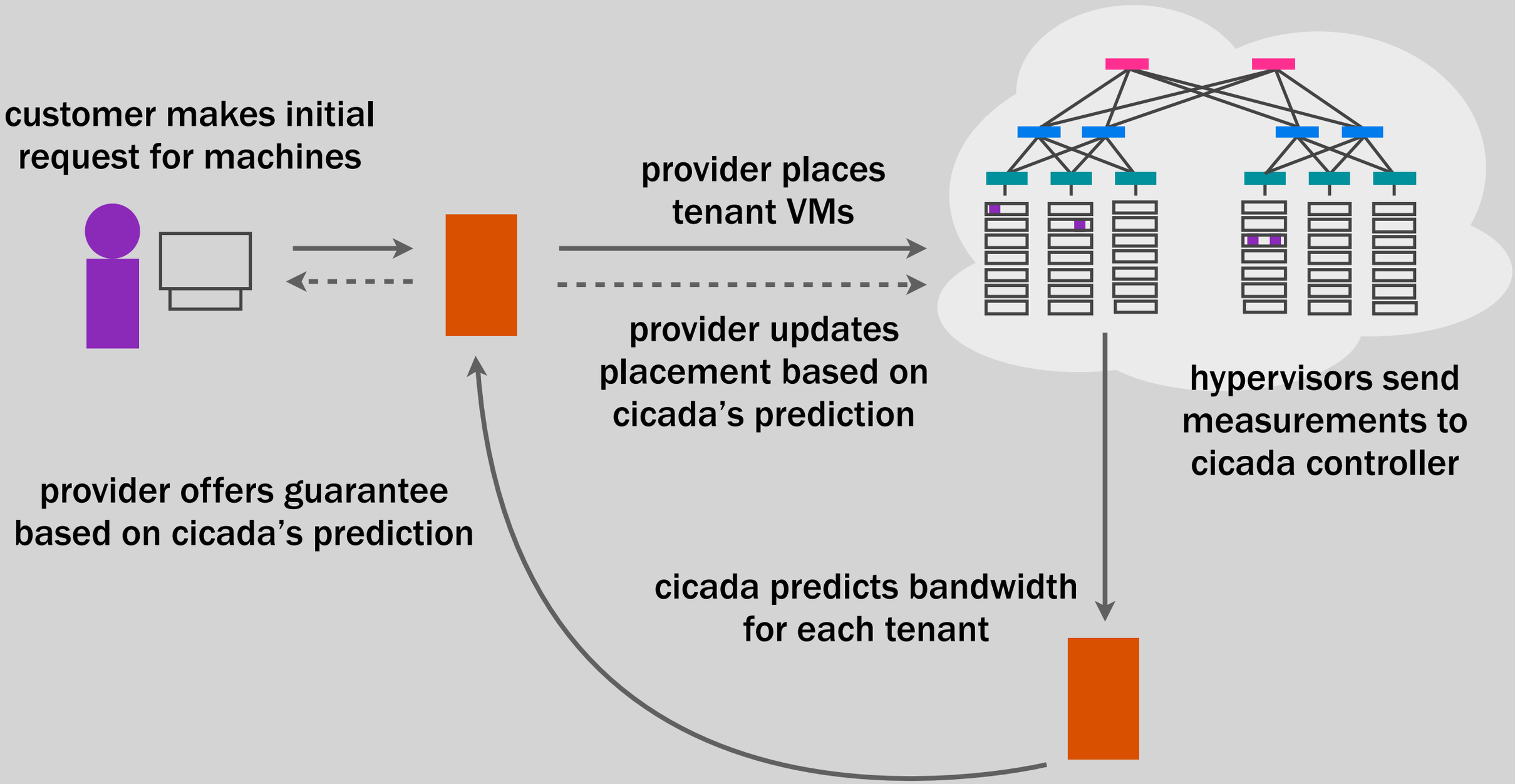
CICADA'S ARCHITECTURE

cicada makes predictions about an application's traffic to automatically generate a guarantee



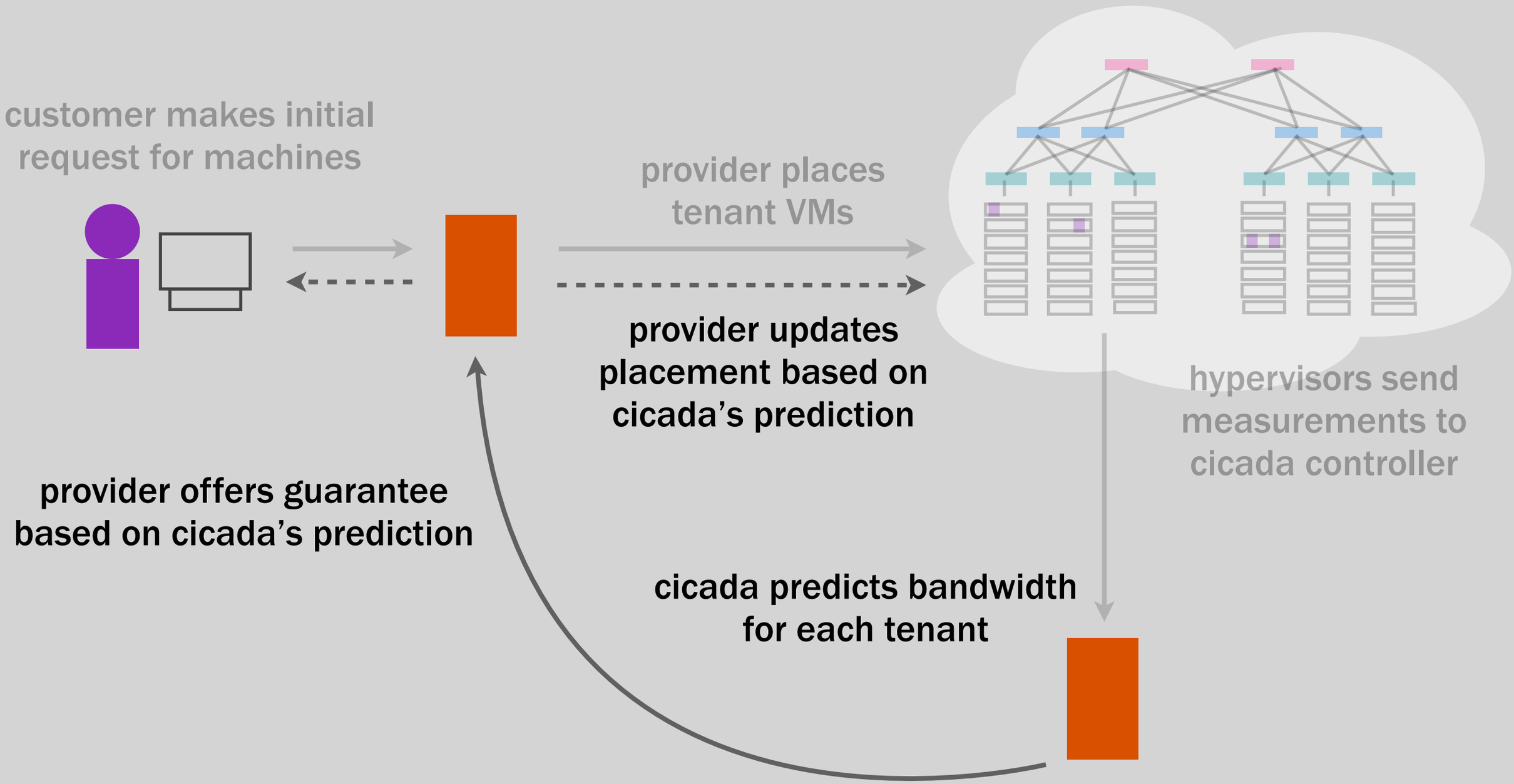
CICADA'S ARCHITECTURE

cicada makes **predictions** about an application's traffic to automatically generate a guarantee



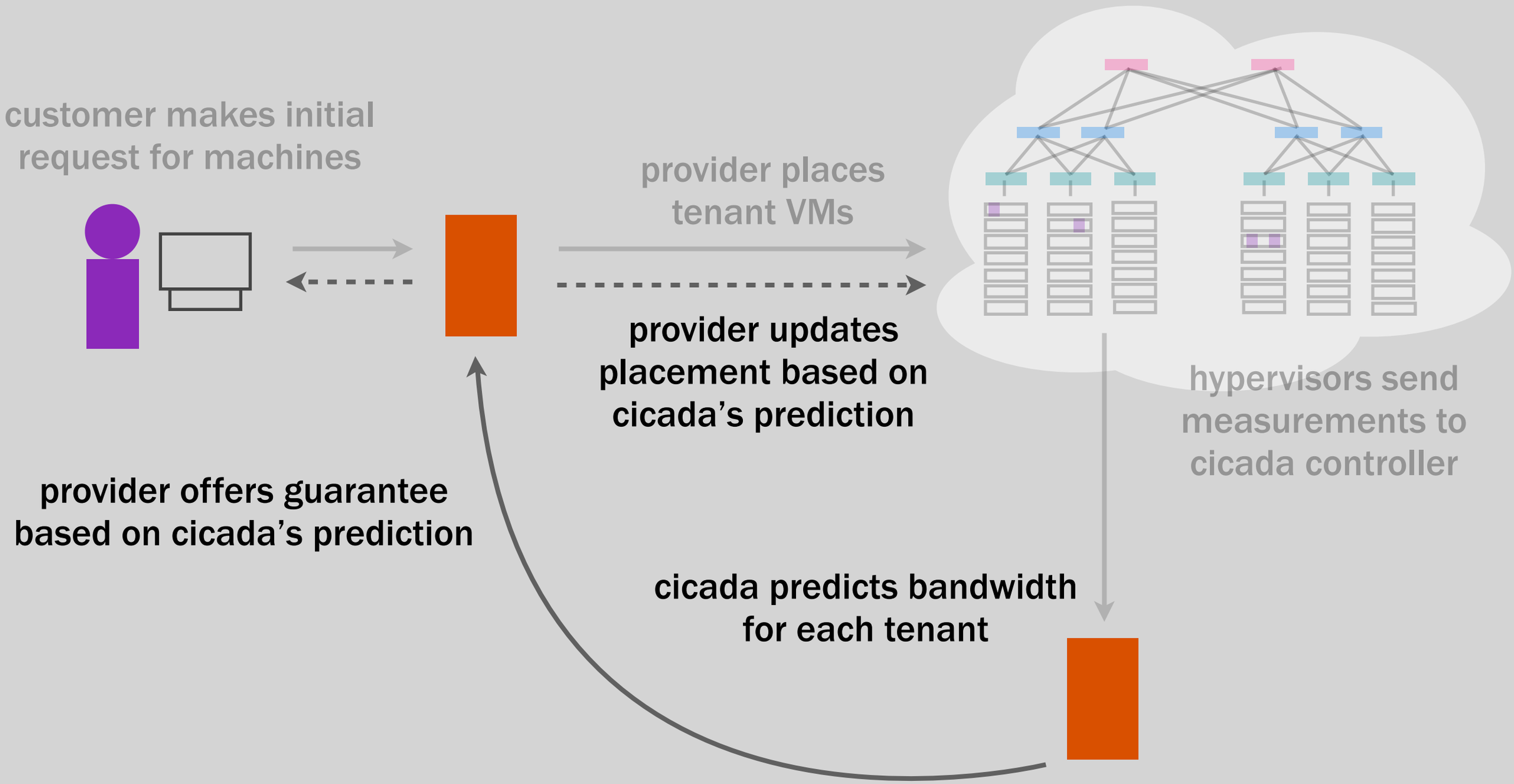
CICADA'S ARCHITECTURE

cicada makes predictions about an application's traffic to automatically generate a guarantee



CICADA'S ARCHITECTURE

problem: is application traffic actually predictable? if so, is it captured by existing models?



APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic

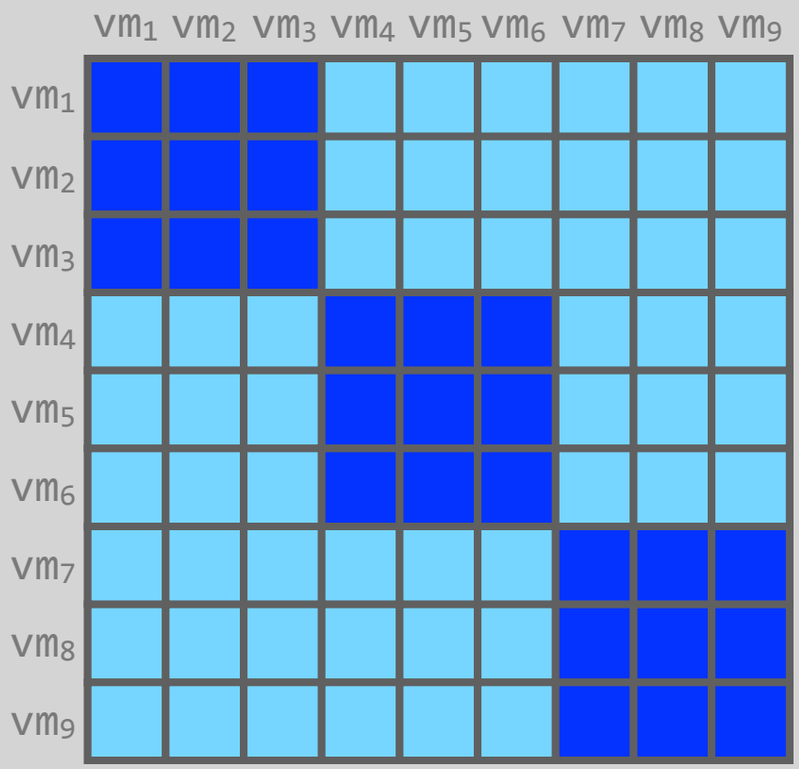
APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic

	vm_1	vm_2	vm_3	vm_4	vm_5	vm_6	vm_7	vm_8	vm_9
vm_1									
vm_2									
vm_3									
vm_4									
vm_5									
vm_6									
vm_7									
vm_8									
vm_9									

APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic



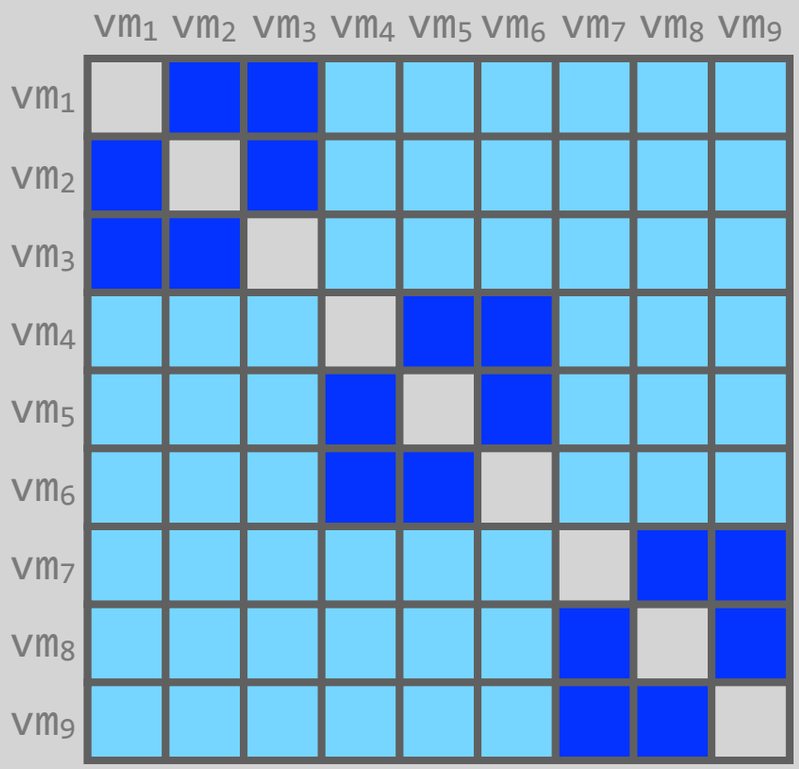
rigid application
(similar to VOC [1])

more data
less data

[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic

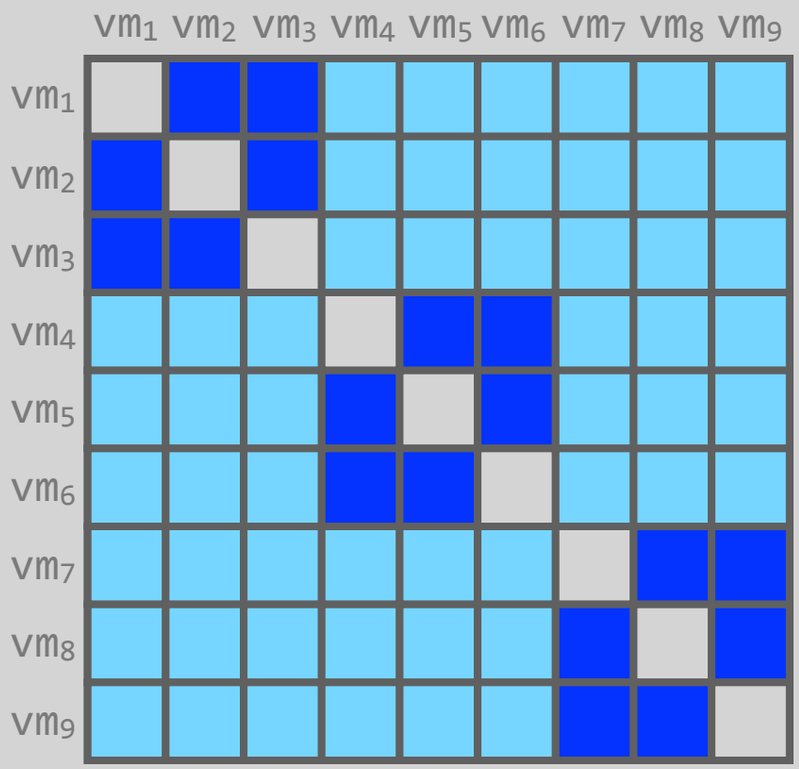


rigid application
(similar to VOC [1])

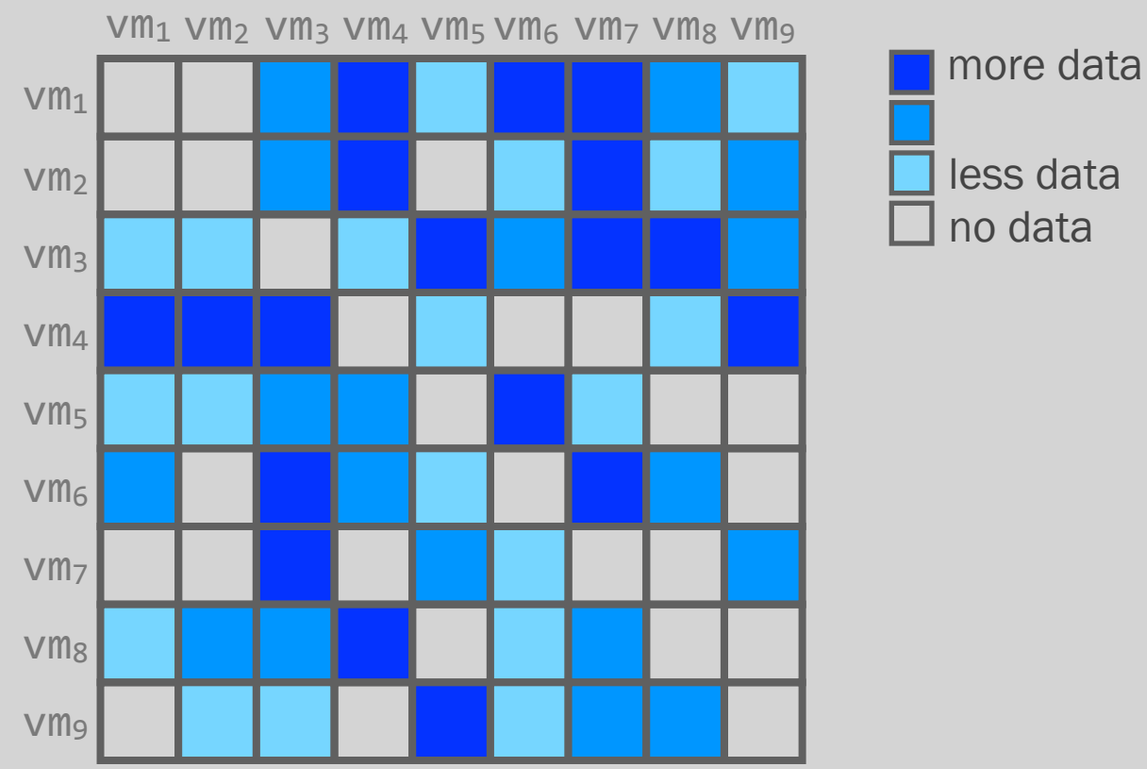
[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic



rigid application
(similar to VOC [1])

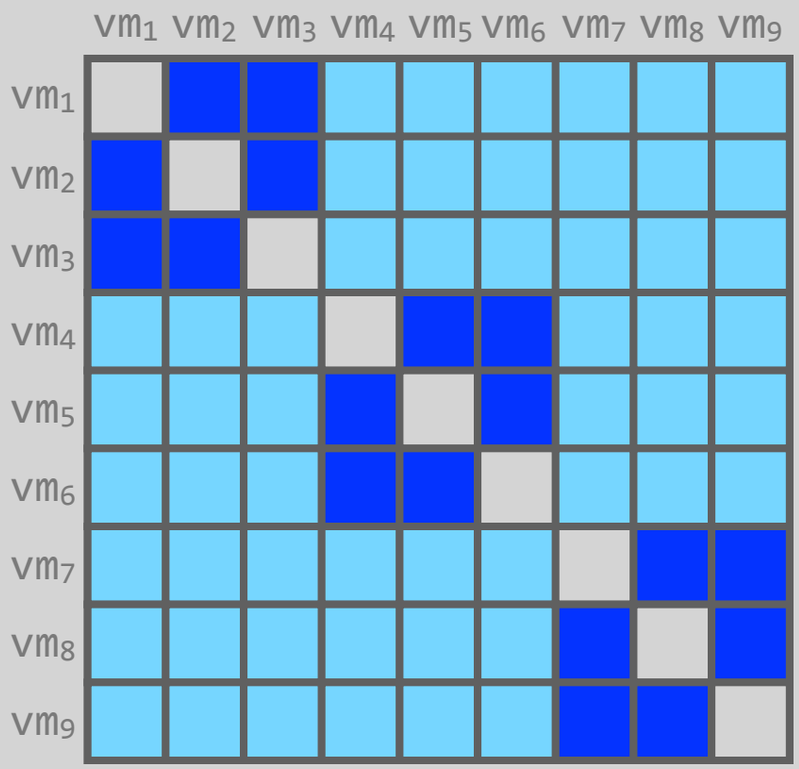


variable application

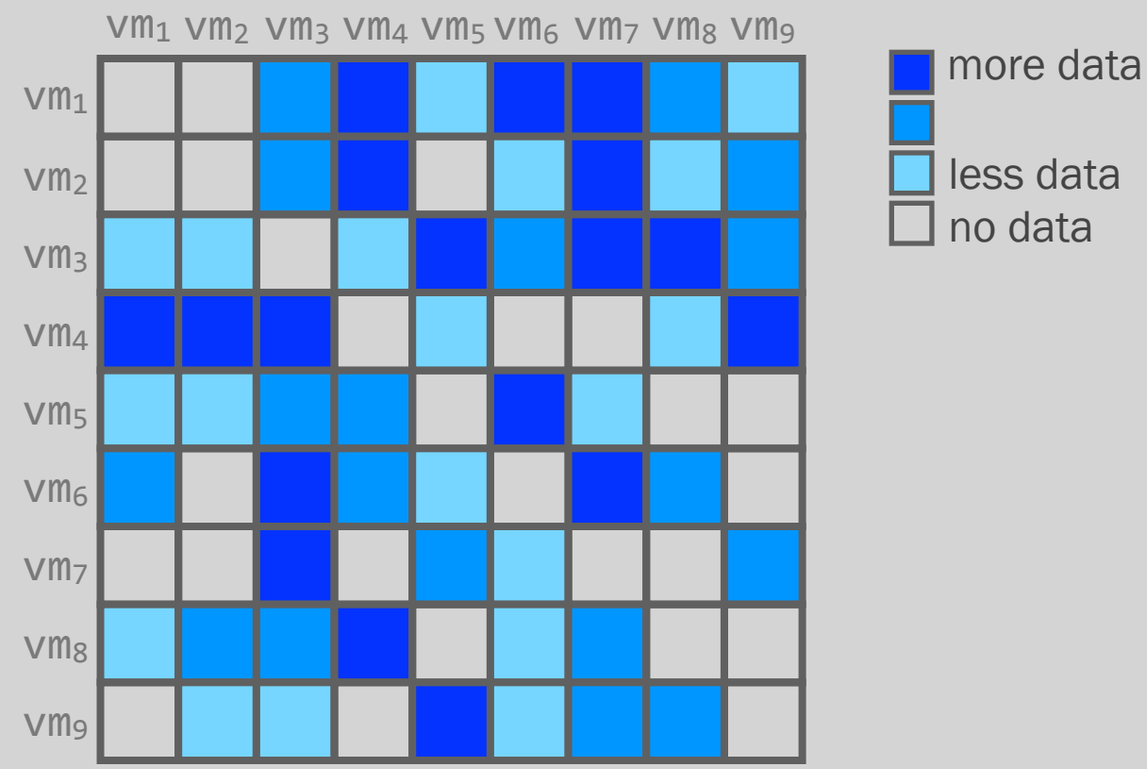
[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic



rigid application
(similar to VOC [1])



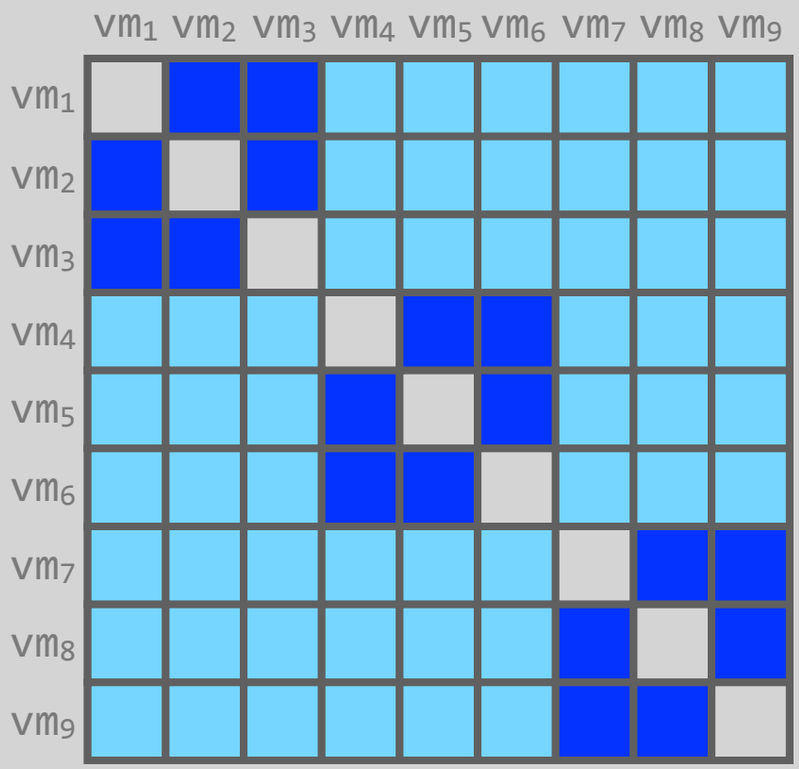
variable application

spatial variability: different pairs of VMs transfer different amounts of data

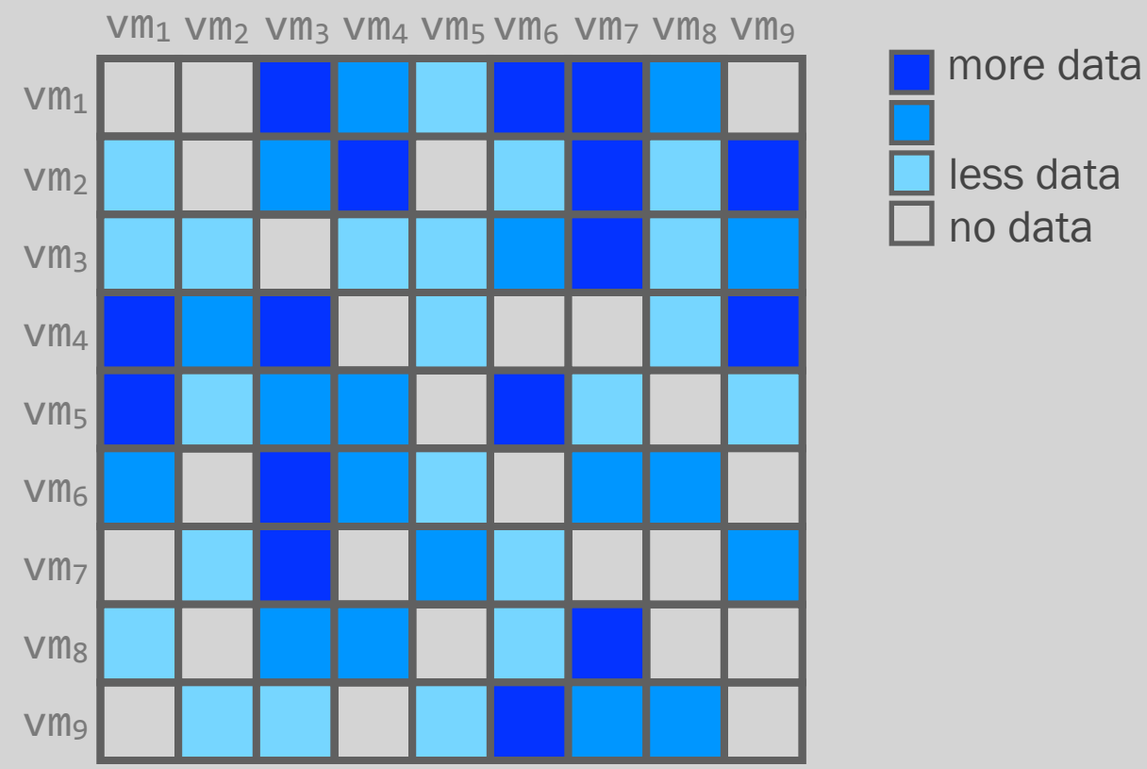
[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic



rigid application
(similar to VOC [1])



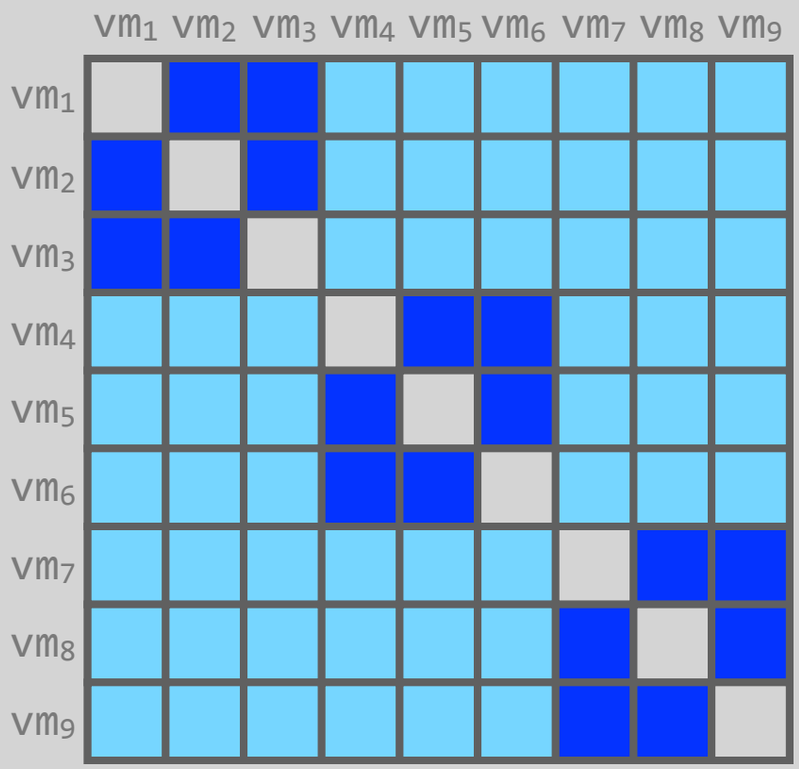
variable application

spatial variability: different pairs of VMs transfer different amounts of data

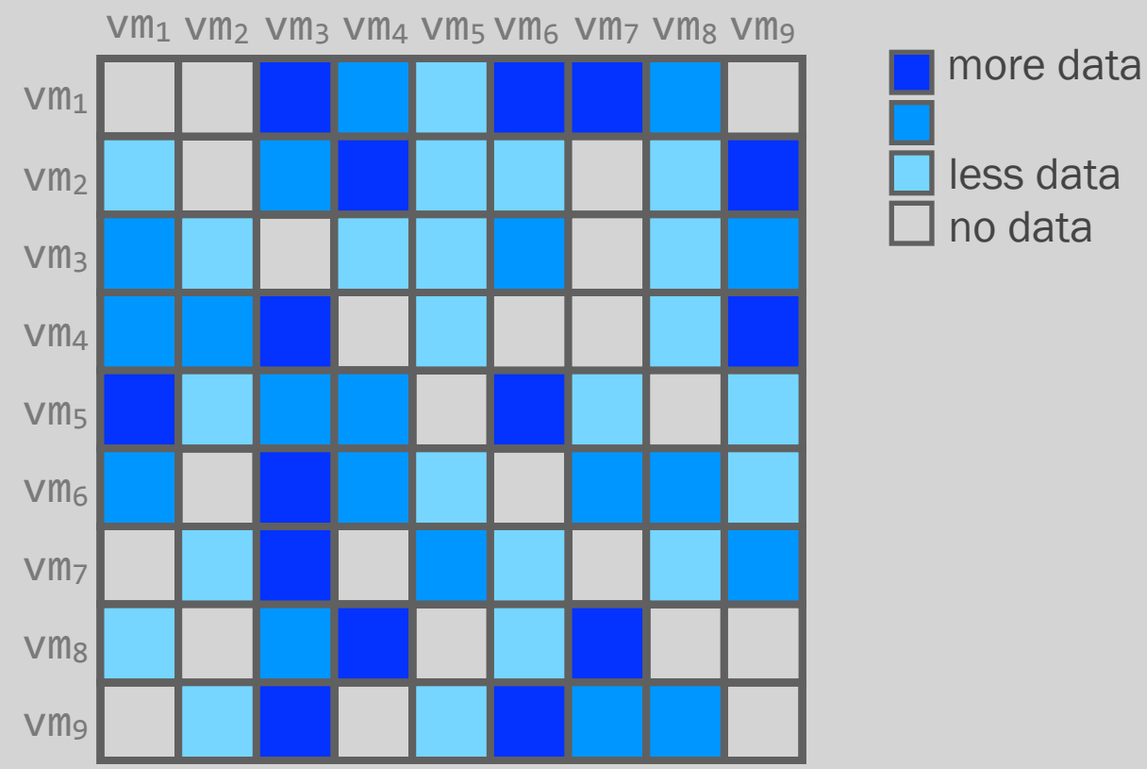
[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic



rigid application
(similar to VOC [1])



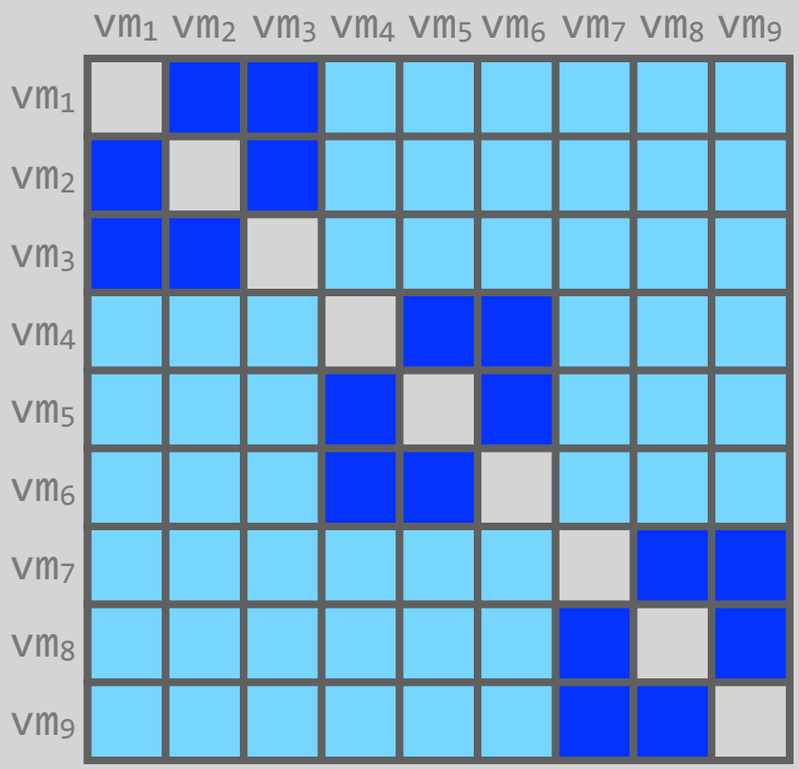
variable application

spatial variability: different pairs of VMs transfer different amounts of data

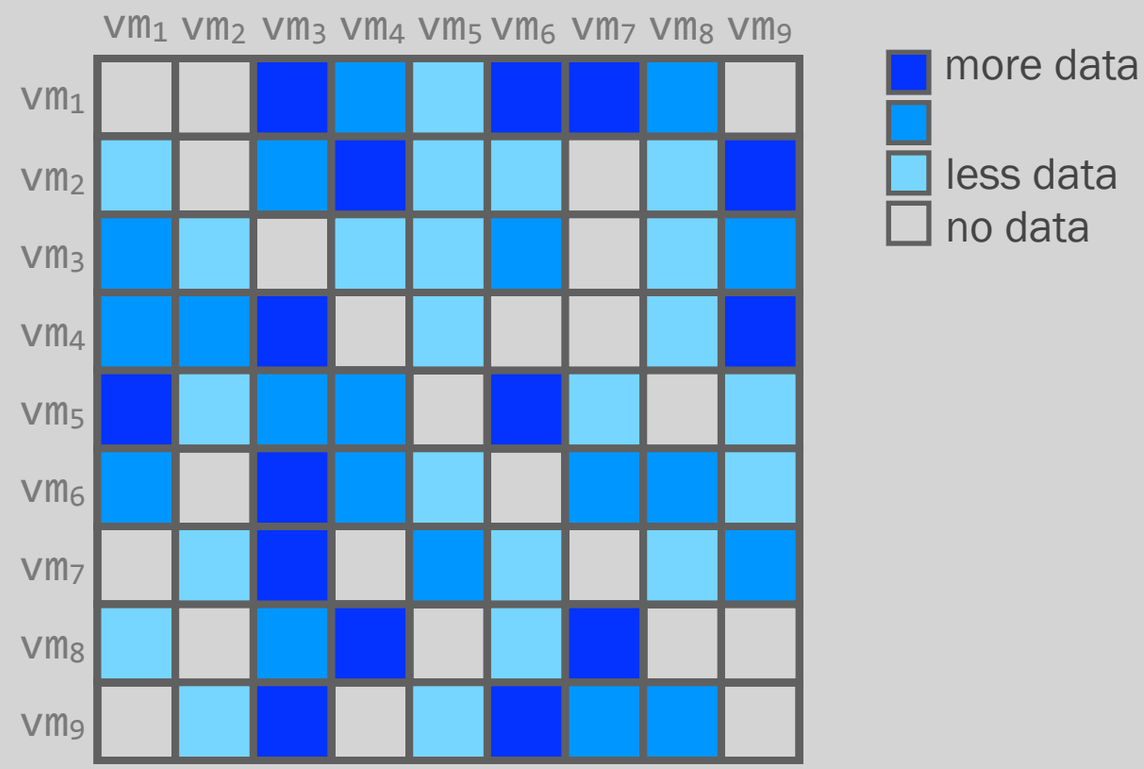
[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

APPLICATION TRAFFIC

to design cicada's prediction algorithm, we need to understand how cloud applications send traffic



rigid application
(similar to VOC [1])



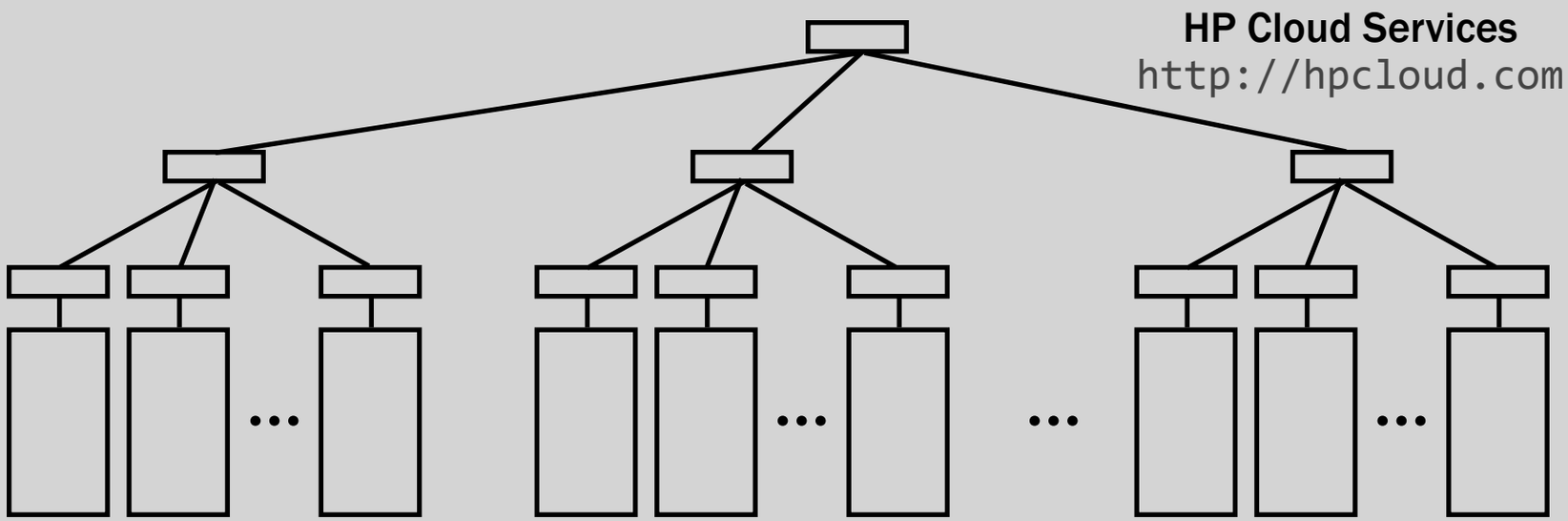
variable application

spatial variability: different pairs of VMs transfer different amounts of data

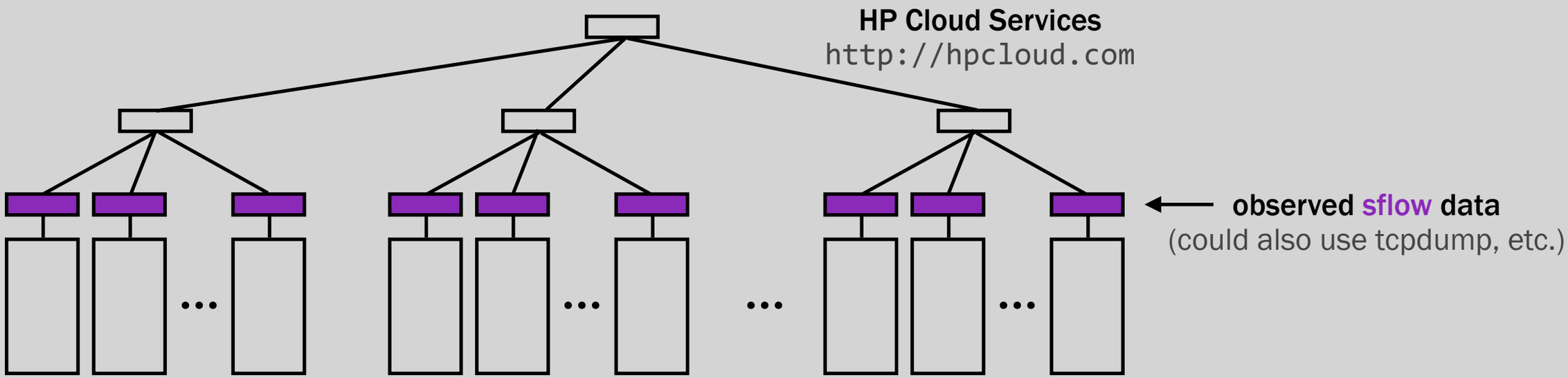
temporal variability: pairs of VMs transfer different amounts of data at different times

[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

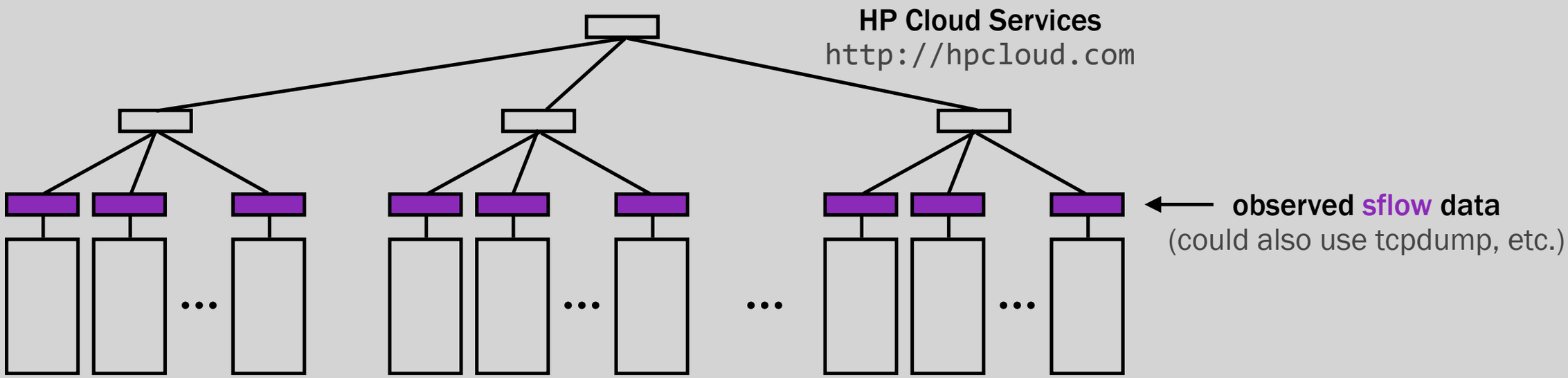
DATASET



DATASET

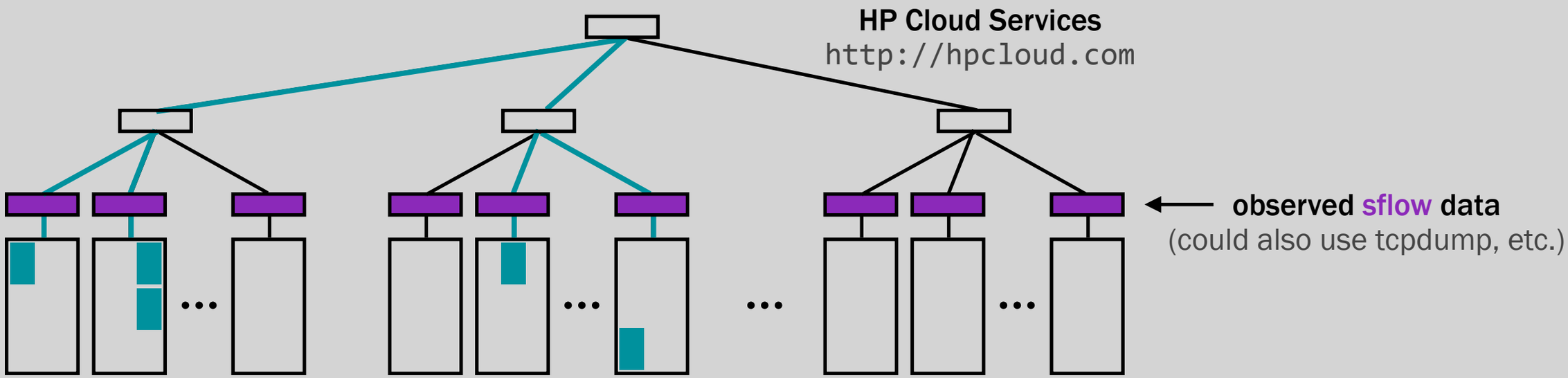

HP Cloud Services
http://hpcloud.com
observed sflow data
(could also use tcpdump, etc.)

DATASET

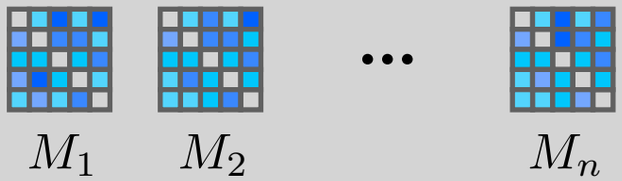


(in this talk)
collected one traffic matrix per hour, for each application

DATASET



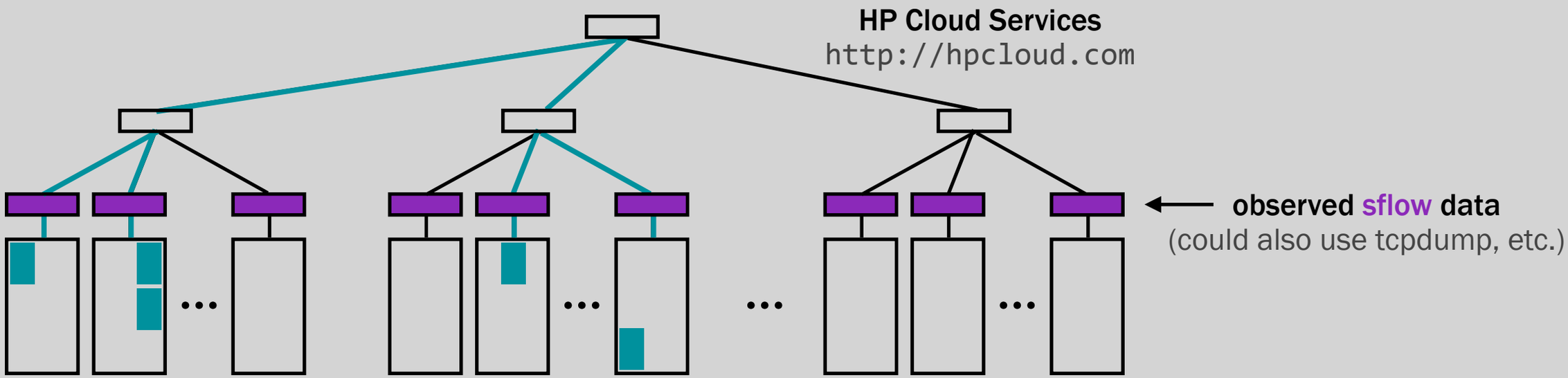
(in this talk)
collected one traffic matrix per hour, for each application



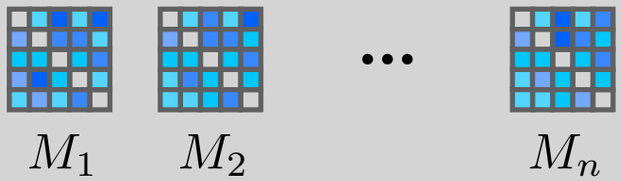
observed traffic

each entry represents the number of bytes transferred between i and j

DATASET



(in this talk)
collected one traffic matrix per hour, for each application



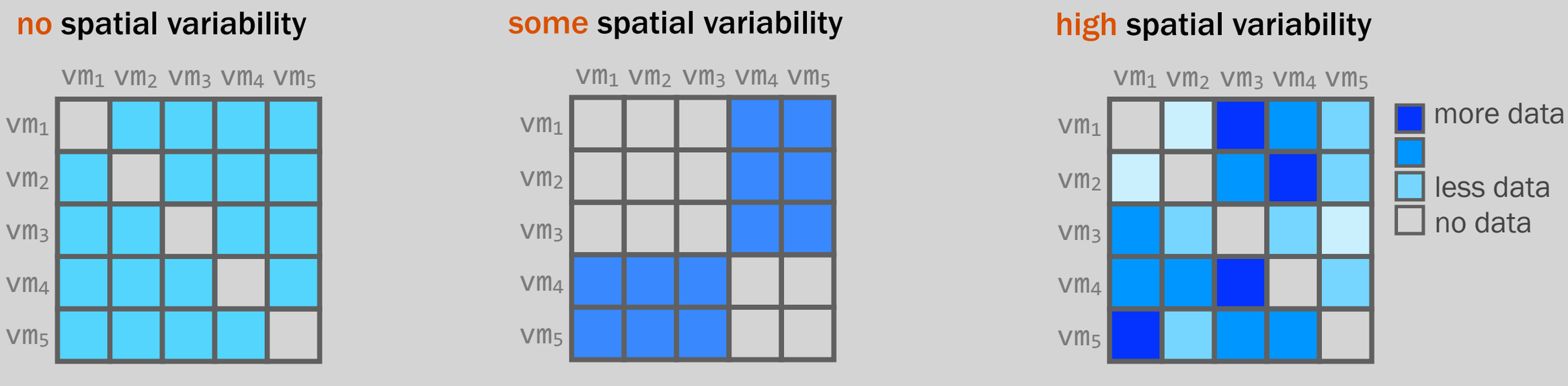
observed traffic

each entry represents the number of bytes transferred between i and j

goal: use this dataset to quantify spatial and temporal variability

SPATIAL VARIABILITY

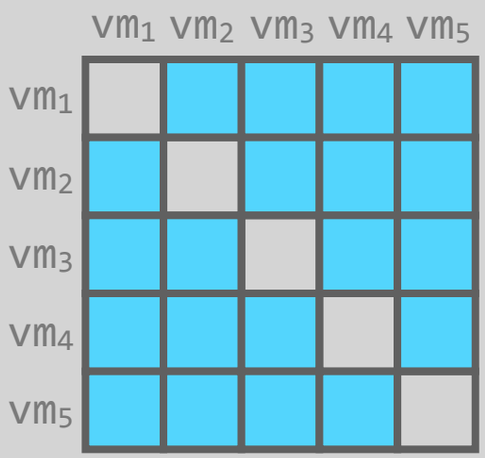
how do we quantify spatial variability?



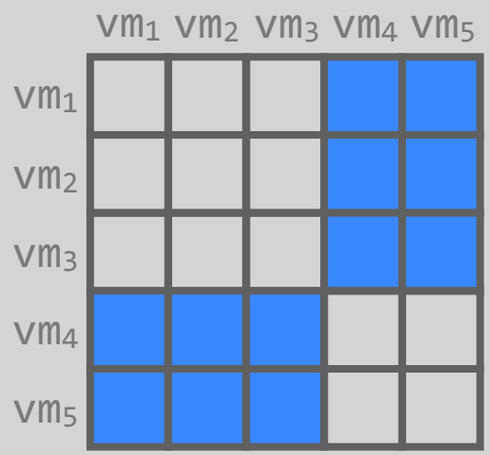
SPATIAL VARIABILITY

how do we quantify spatial variability?

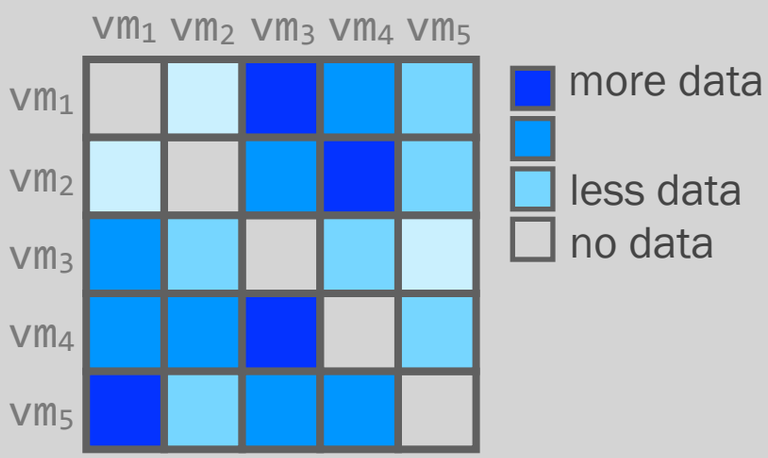
no spatial variability



some spatial variability



high spatial variability

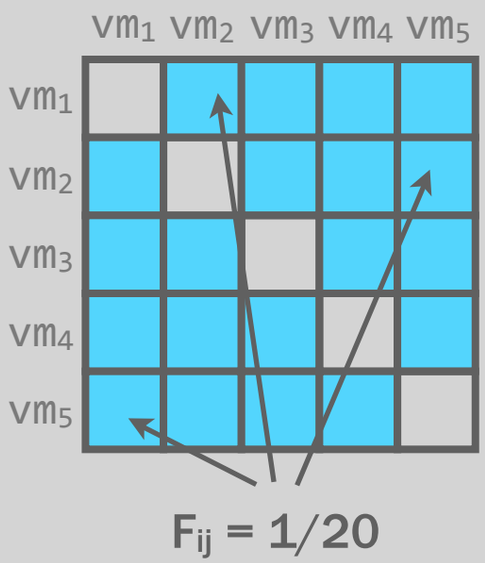


1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

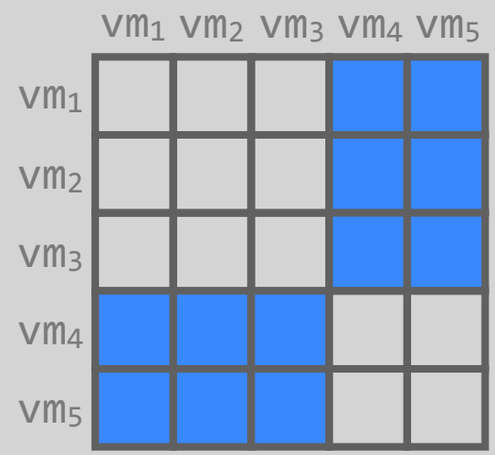
SPATIAL VARIABILITY

how do we quantify spatial variability?

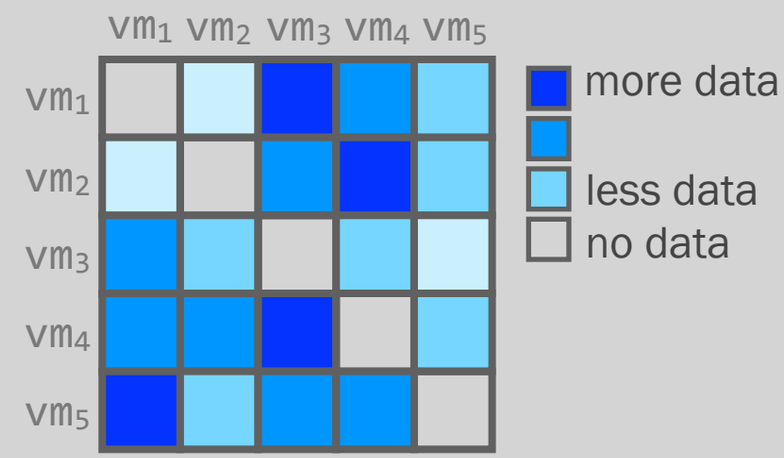
no spatial variability



some spatial variability



high spatial variability

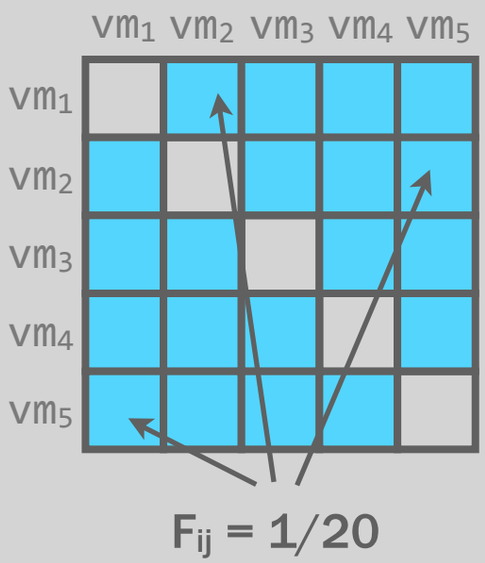


1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

SPATIAL VARIABILITY

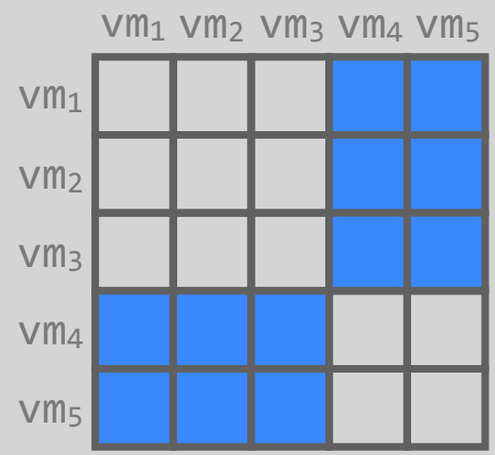
how do we quantify spatial variability?

no spatial variability

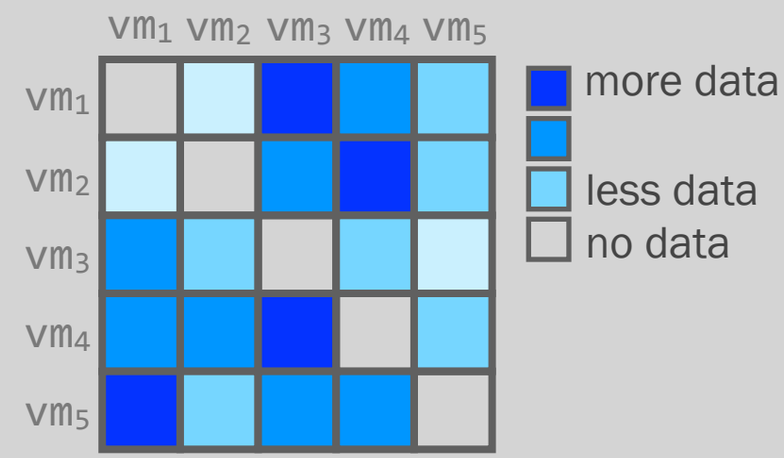


all F_{ij} values are equal

some spatial variability



high spatial variability

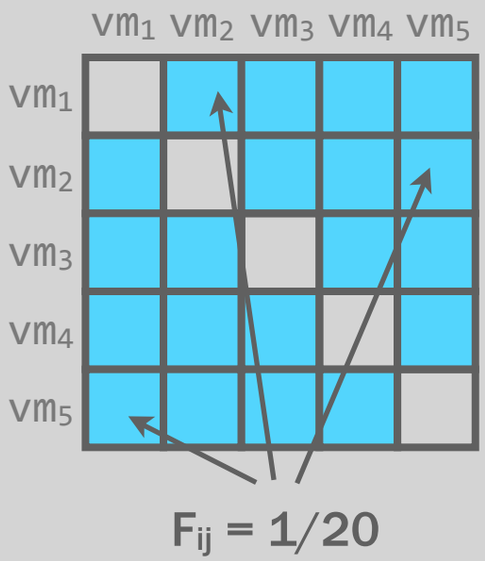


1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

SPATIAL VARIABILITY

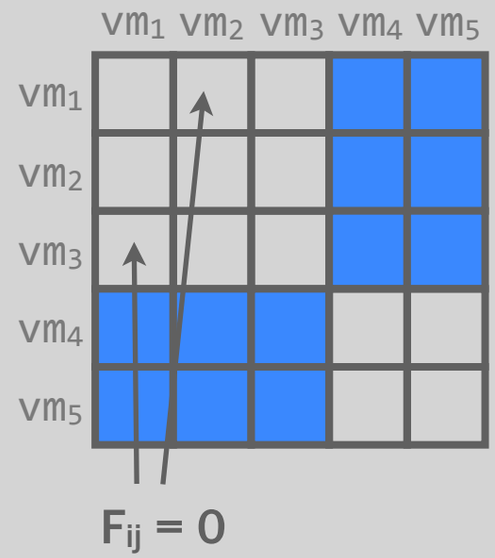
how do we quantify spatial variability?

no spatial variability

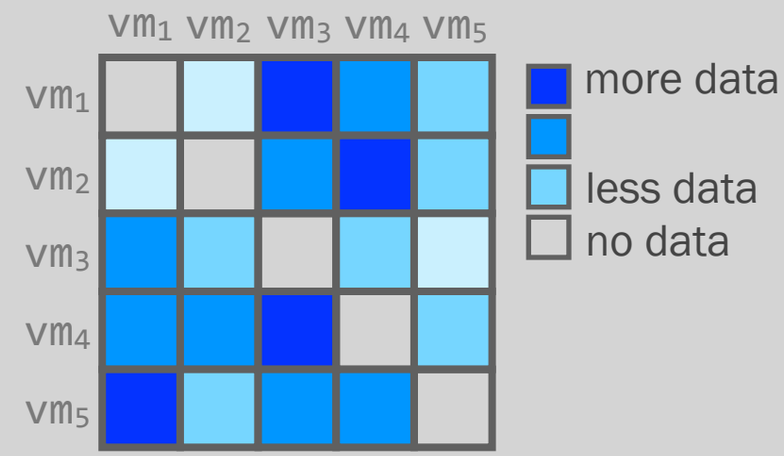


all F_{ij} values are equal

some spatial variability



high spatial variability

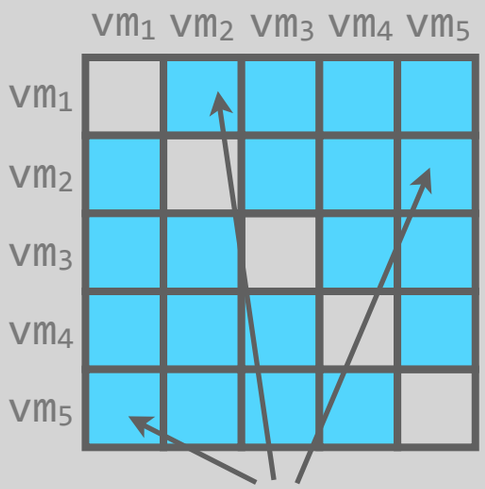


1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

SPATIAL VARIABILITY

how do we quantify spatial variability?

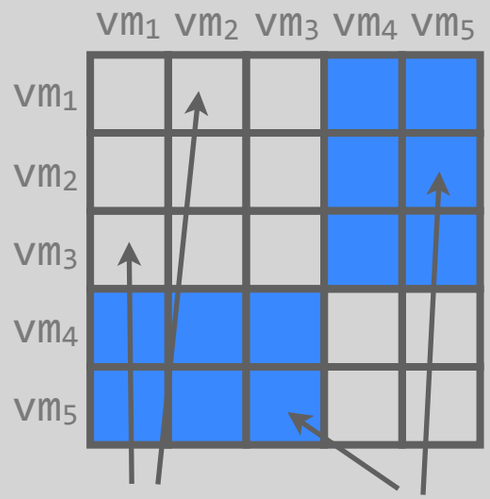
no spatial variability



$F_{ij} = 1/20$

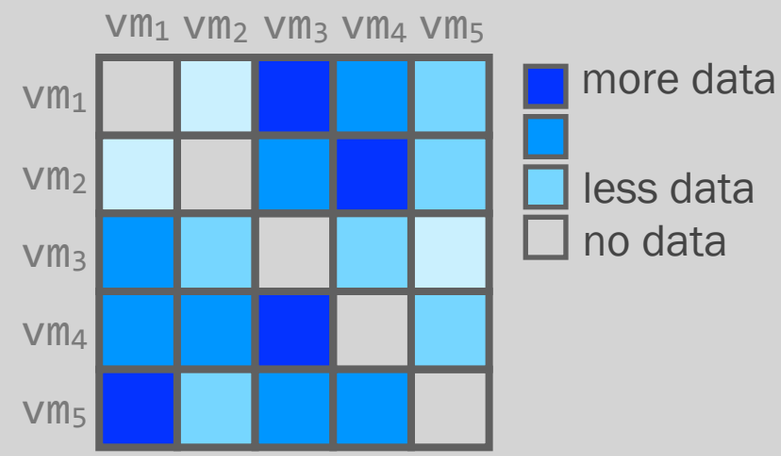
all F_{ij} values are equal

some spatial variability



$F_{ij} = 0$

high spatial variability



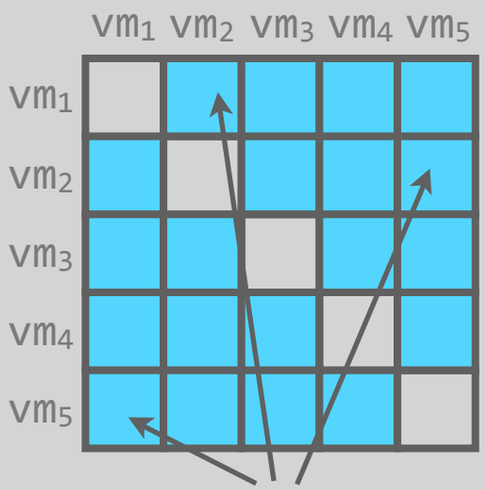
$F_{ij} = 1/12$

1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

SPATIAL VARIABILITY

how do we quantify spatial variability?

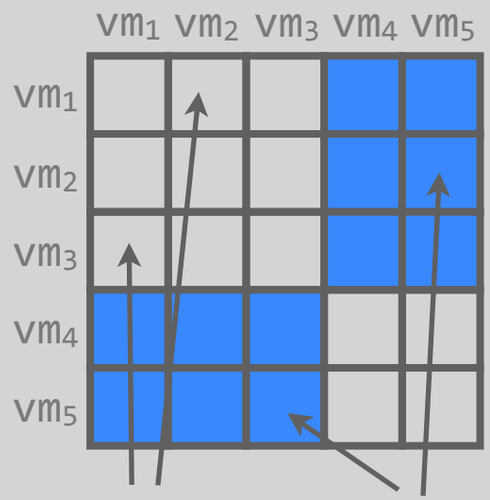
no spatial variability



$F_{ij} = 1/20$

all F_{ij} values are equal

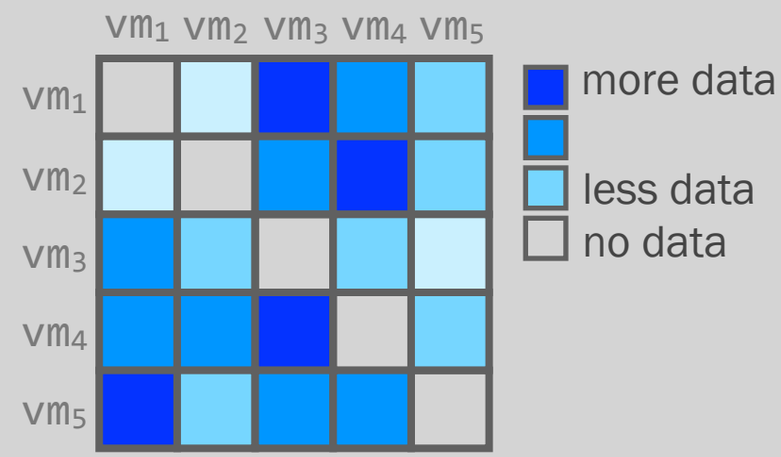
some spatial variability



$F_{ij} = 0$ $F_{ij} = 1/12$

two distinct F_{ij} values

high spatial variability

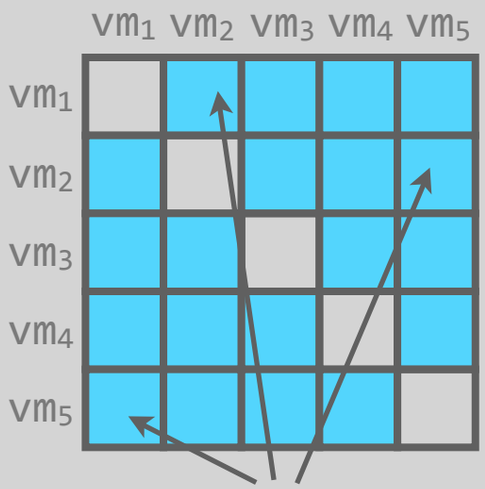


1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

SPATIAL VARIABILITY

how do we quantify spatial variability?

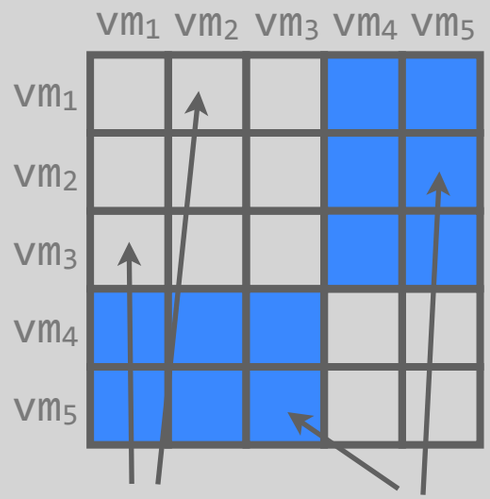
no spatial variability



$F_{ij} = 1/20$

all F_{ij} values are equal

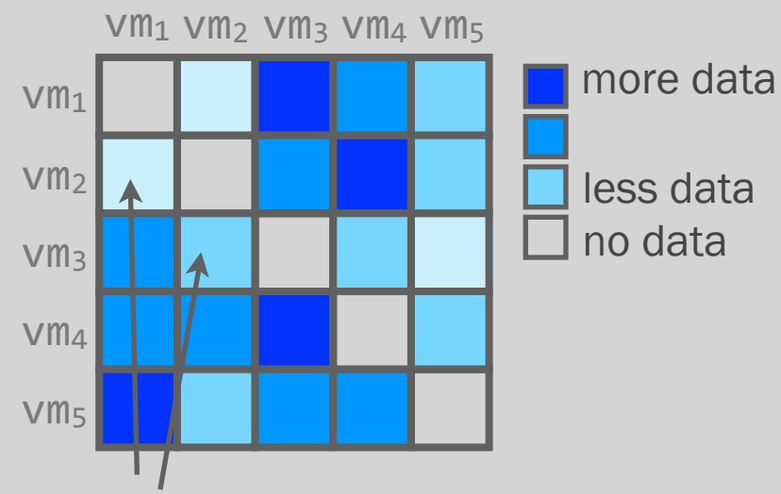
some spatial variability



$F_{ij} = 0$ $F_{ij} = 1/12$

two distinct F_{ij} values

high spatial variability



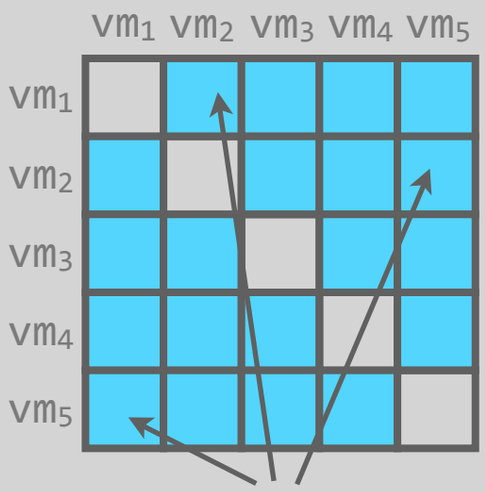
$F_{ij} < 1/20$

1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

SPATIAL VARIABILITY

how do we quantify spatial variability?

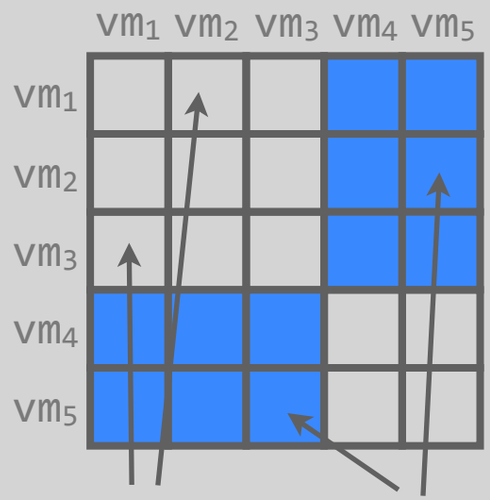
no spatial variability



$F_{ij} = 1/20$

all F_{ij} values are equal

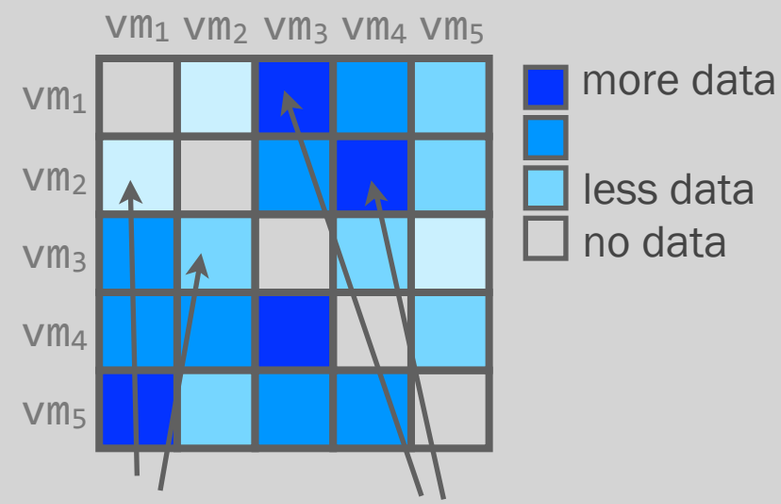
some spatial variability



$F_{ij} = 0$ $F_{ij} = 1/12$

two distinct F_{ij} values

high spatial variability



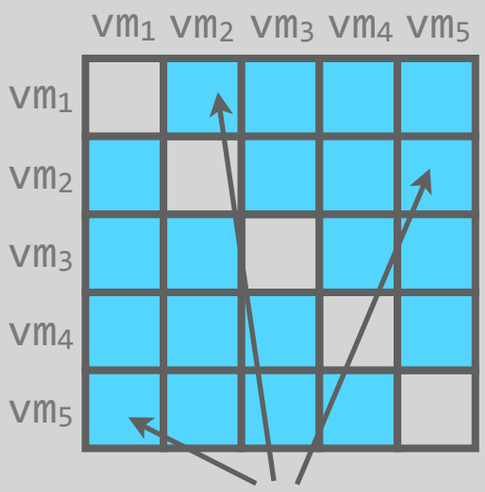
$F_{ij} < 1/20$ $F_{ij} > 1/20$

1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

SPATIAL VARIABILITY

how do we quantify spatial variability?

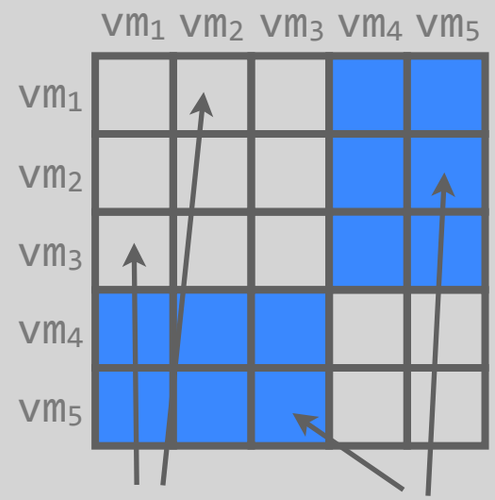
no spatial variability



$F_{ij} = 1/20$

all F_{ij} values are equal

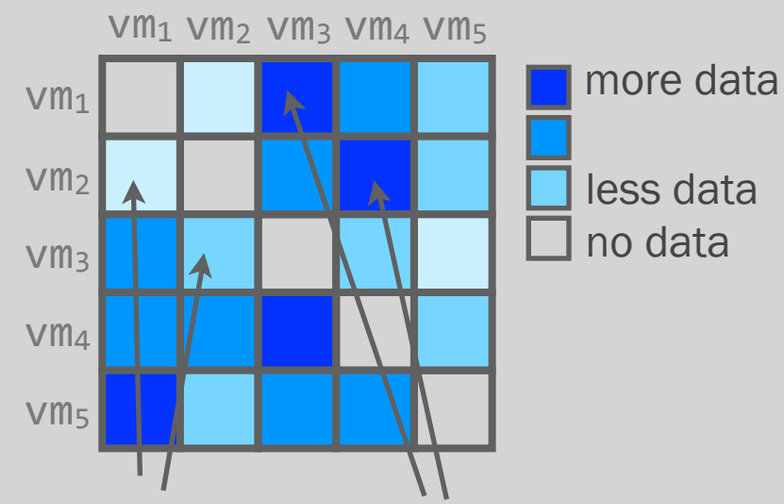
some spatial variability



$F_{ij} = 0$ $F_{ij} = 1/12$

two distinct F_{ij} values

high spatial variability



$F_{ij} < 1/20$ $F_{ij} > 1/20$

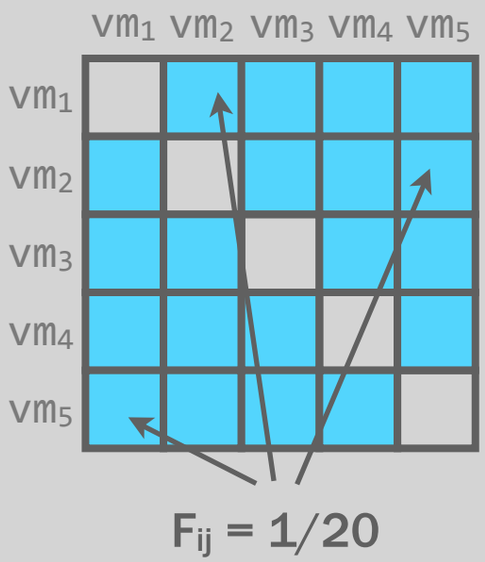
many distinct F_{ij} values

1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j

SPATIAL VARIABILITY

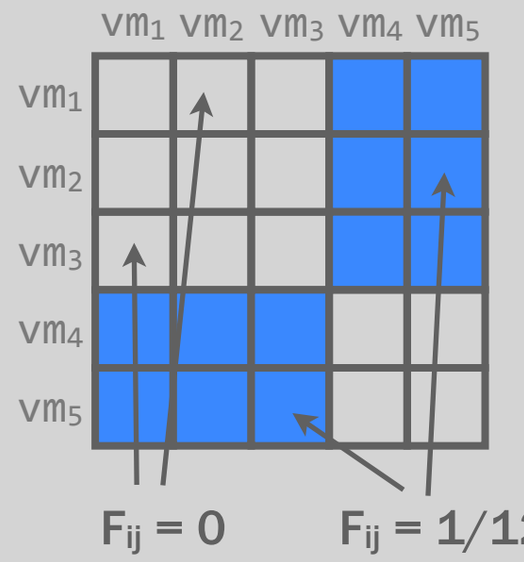
how do we quantify spatial variability?

no spatial variability



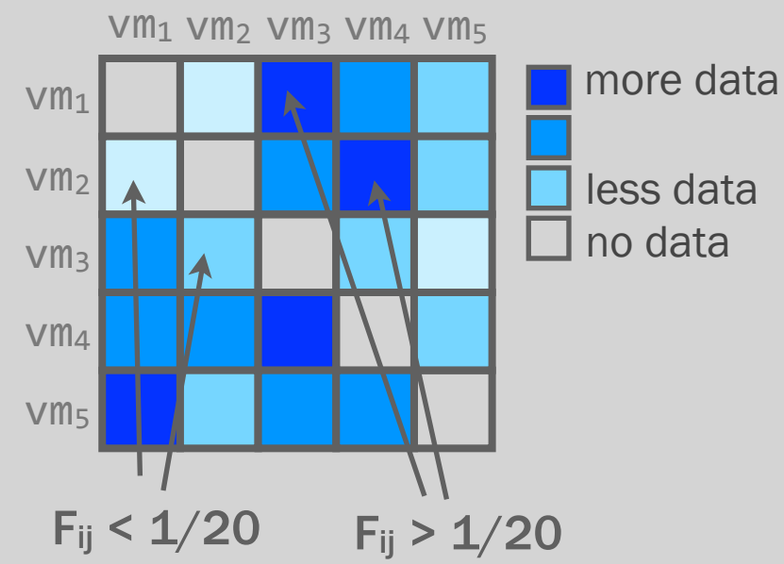
all F_{ij} values are equal

some spatial variability



two distinct F_{ij} values

high spatial variability



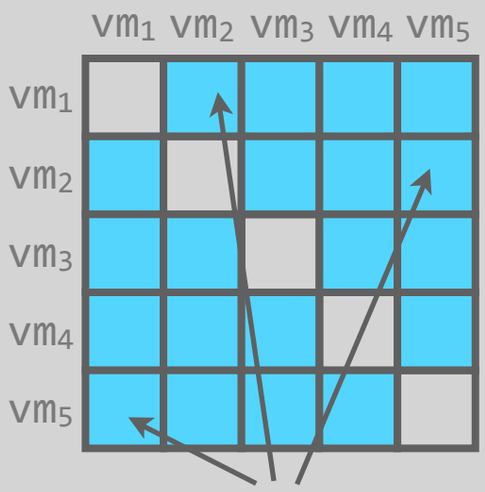
many distinct F_{ij} values

1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j
2. calculate the coefficient of variation (σ/μ) of the F_{ij} values

SPATIAL VARIABILITY

how do we quantify spatial variability?

no spatial variability

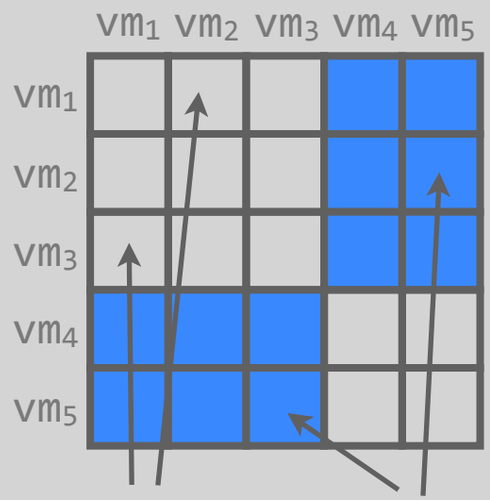


$F_{ij} = 1/20$

all F_{ij} values are equal

$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 0$

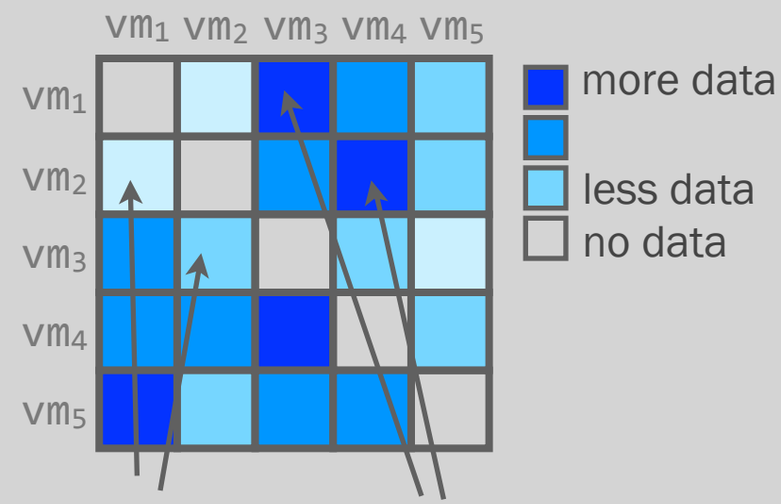
some spatial variability



$F_{ij} = 0$ $F_{ij} = 1/12$

two distinct F_{ij} values

high spatial variability



$F_{ij} < 1/20$ $F_{ij} > 1/20$

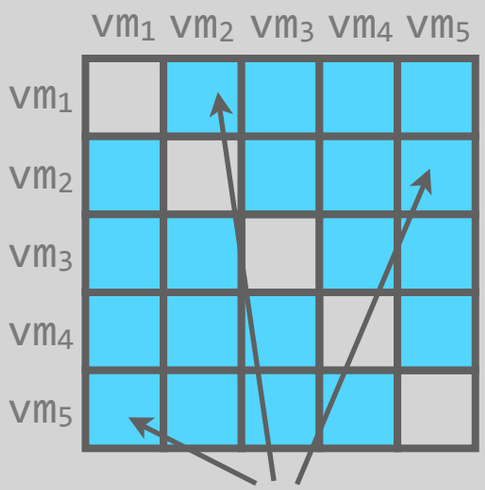
many distinct F_{ij} values

1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j
2. calculate the coefficient of variation (σ/μ) of the F_{ij} values

SPATIAL VARIABILITY

how do we quantify spatial variability?

no spatial variability

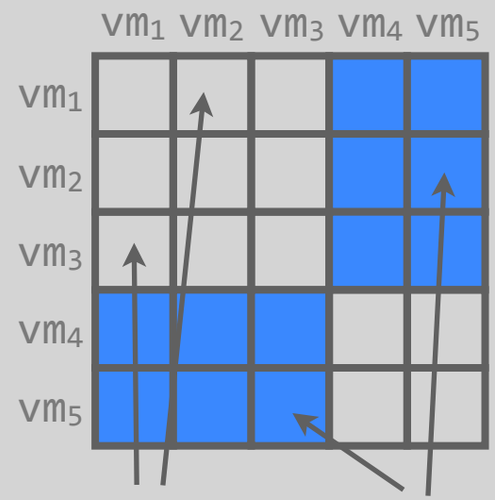


$F_{ij} = 1/20$

all F_{ij} values are equal

$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 0$

some spatial variability

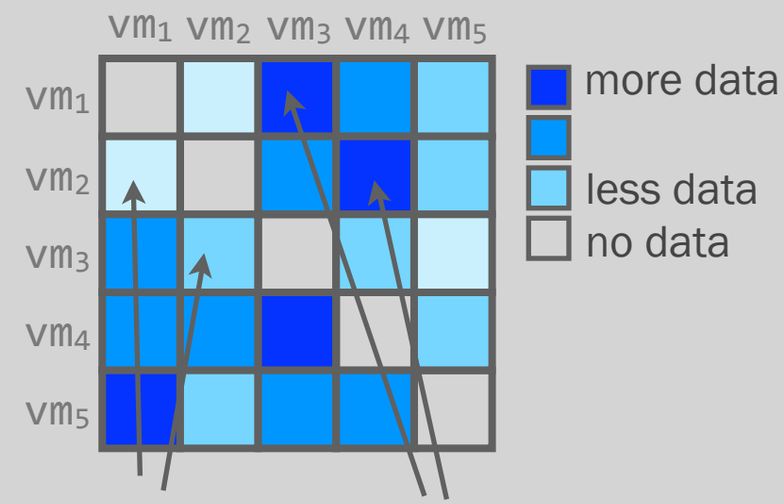


$F_{ij} = 0$ $F_{ij} = 1/12$

two distinct F_{ij} values

$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 1^*$

high spatial variability



$F_{ij} < 1/20$ $F_{ij} > 1/20$

many distinct F_{ij} values

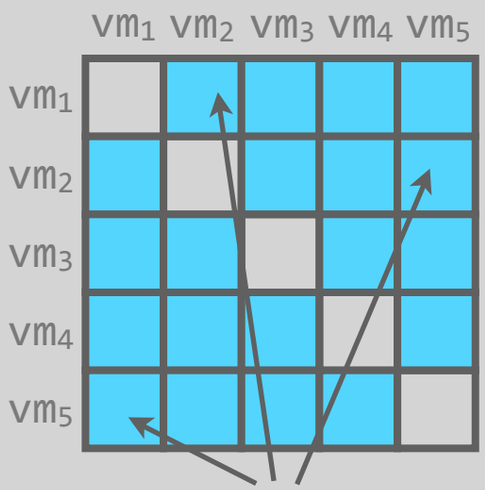
1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j
2. calculate the coefficient of variation (σ/μ) of the F_{ij} values

*see paper for this result

SPATIAL VARIABILITY

how do we quantify spatial variability?

no spatial variability

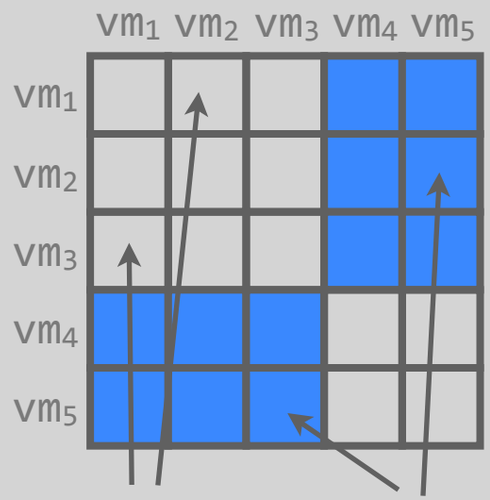


$F_{ij} = 1/20$

all F_{ij} values are equal

$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 0$

some spatial variability

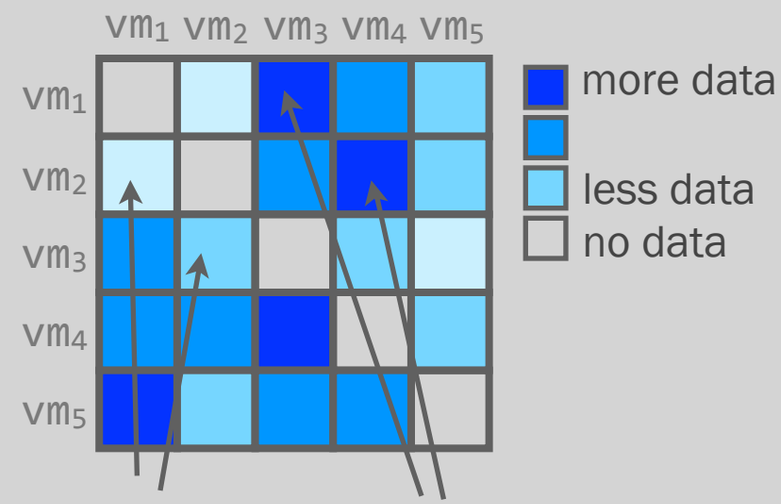


$F_{ij} = 0$ $F_{ij} = 1/12$

two distinct F_{ij} values

$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 1^*$

high spatial variability



$F_{ij} < 1/20$ $F_{ij} > 1/20$

many distinct F_{ij} values

$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) > 1$

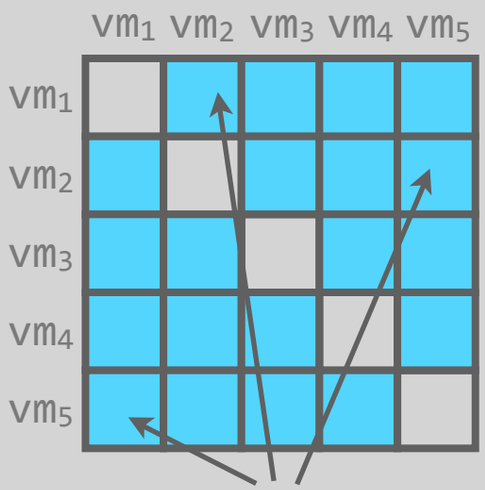
1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j
2. calculate the coefficient of variation (σ/μ) of the F_{ij} values

*see paper for this result

SPATIAL VARIABILITY

how do we quantify spatial variability?

no spatial variability

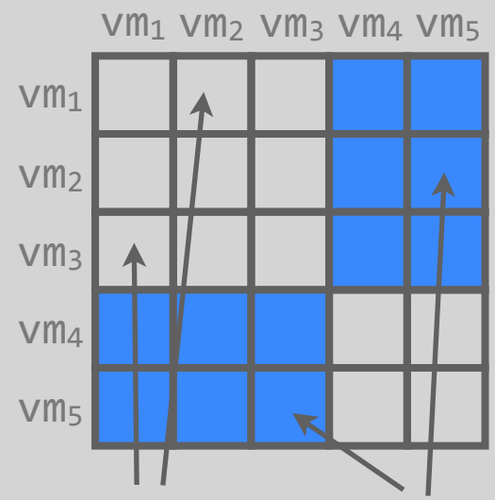


$F_{ij} = 1/20$

all F_{ij} values are equal

$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 0$

some spatial variability

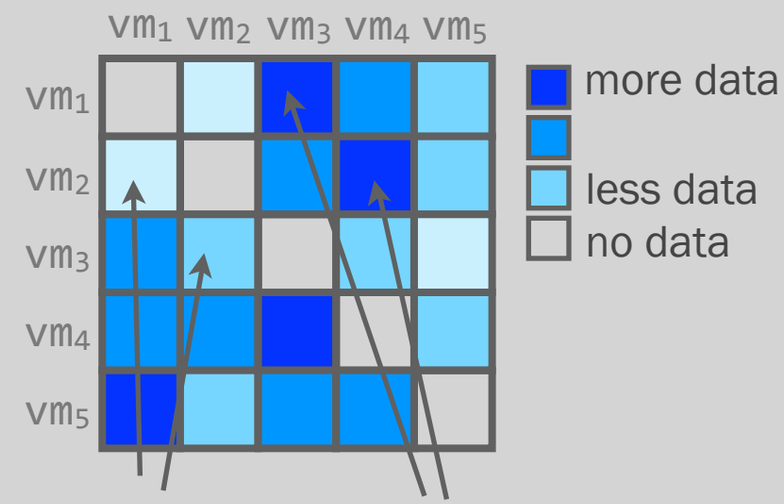


$F_{ij} = 0$ $F_{ij} = 1/12$

two distinct F_{ij} values

$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 1^*$

high spatial variability



$F_{ij} < 1/20$ $F_{ij} > 1/20$

many distinct F_{ij} values

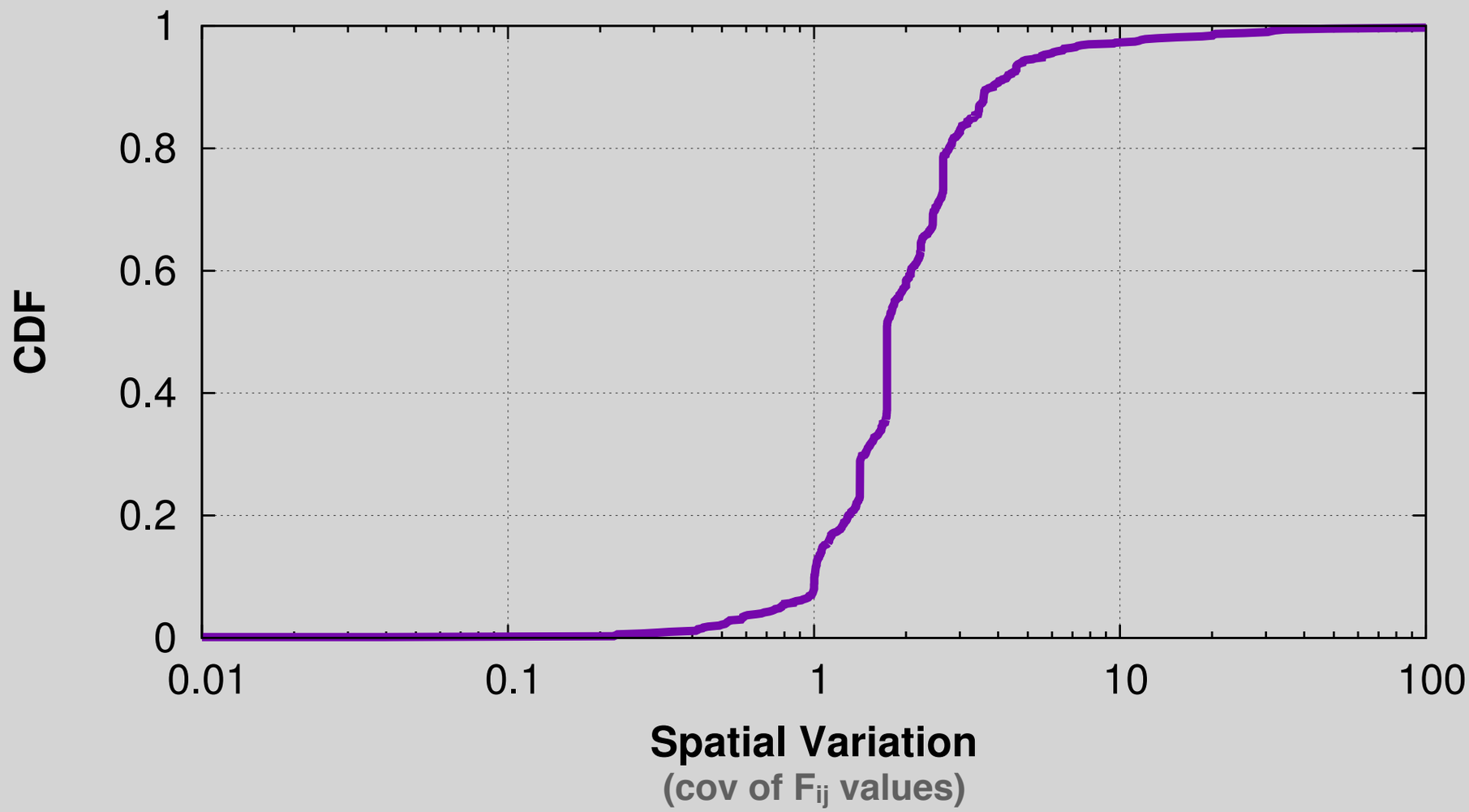
$\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) > 1$

1. let F_{ij} = fraction of tenant traffic sent from VM_i to VM_j
2. calculate the coefficient of variation (σ/μ) of the F_{ij} values

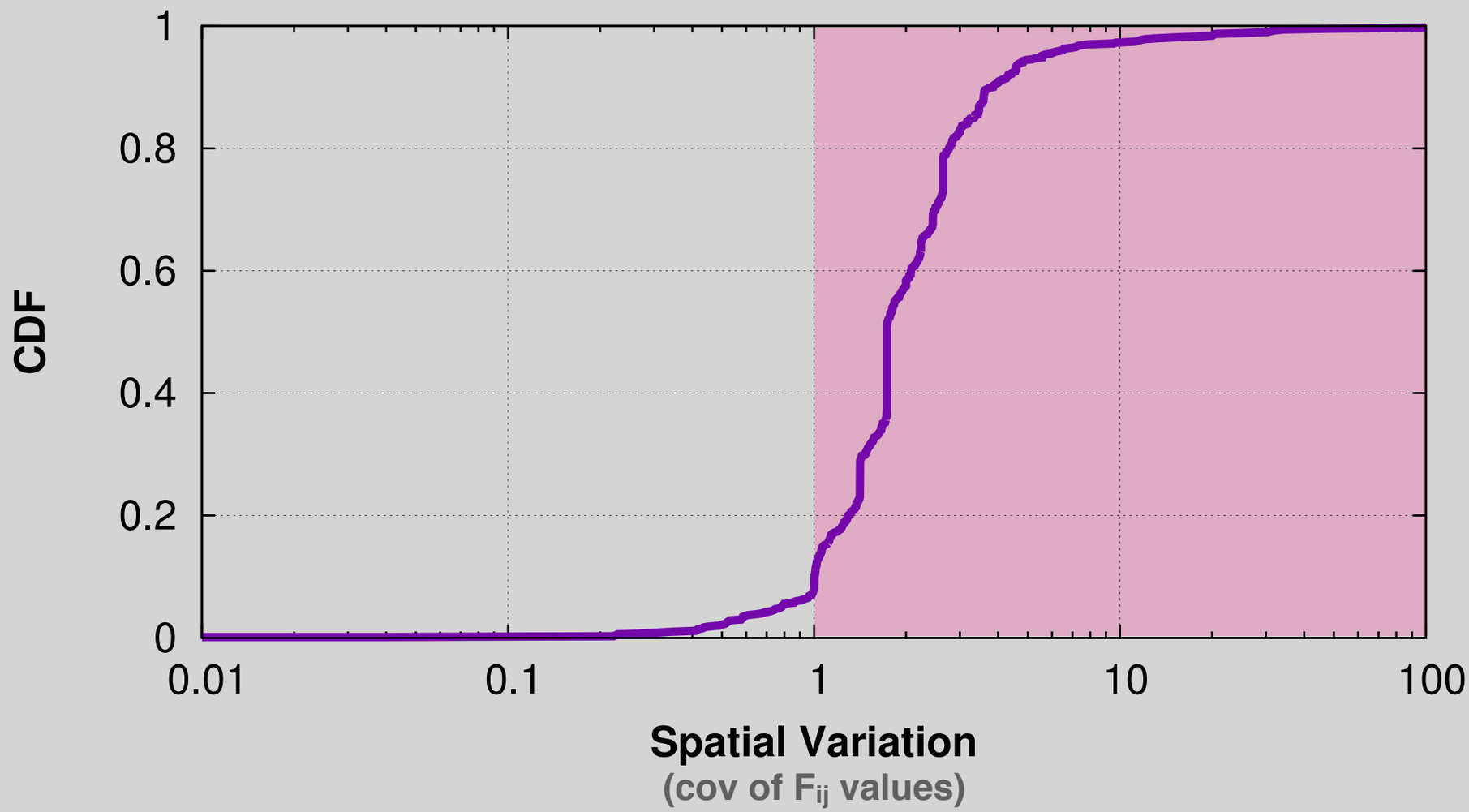
the higher the cov, the higher the spatial variation

*see paper for this result

SPATIAL VARIABILITY

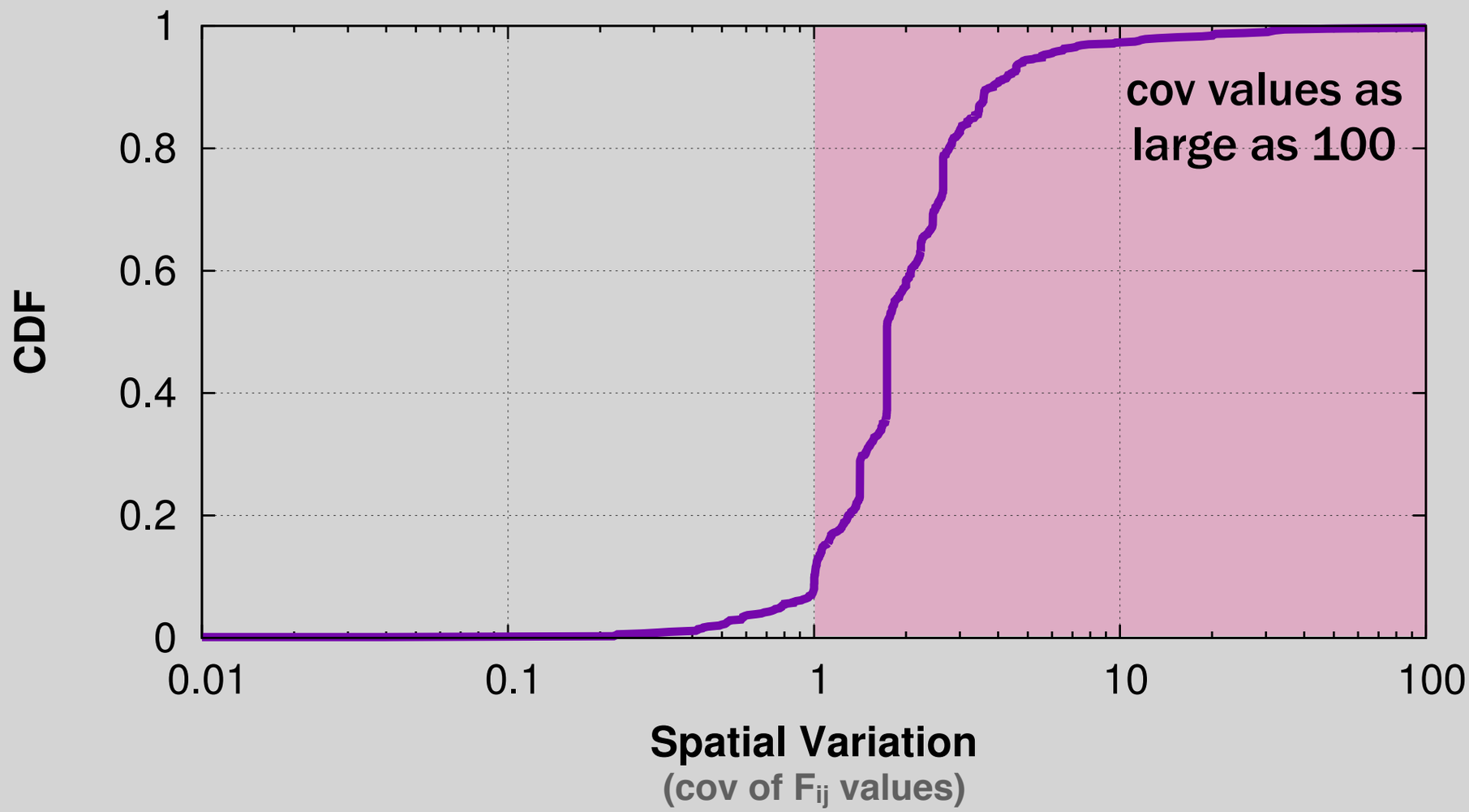


SPATIAL VARIABILITY



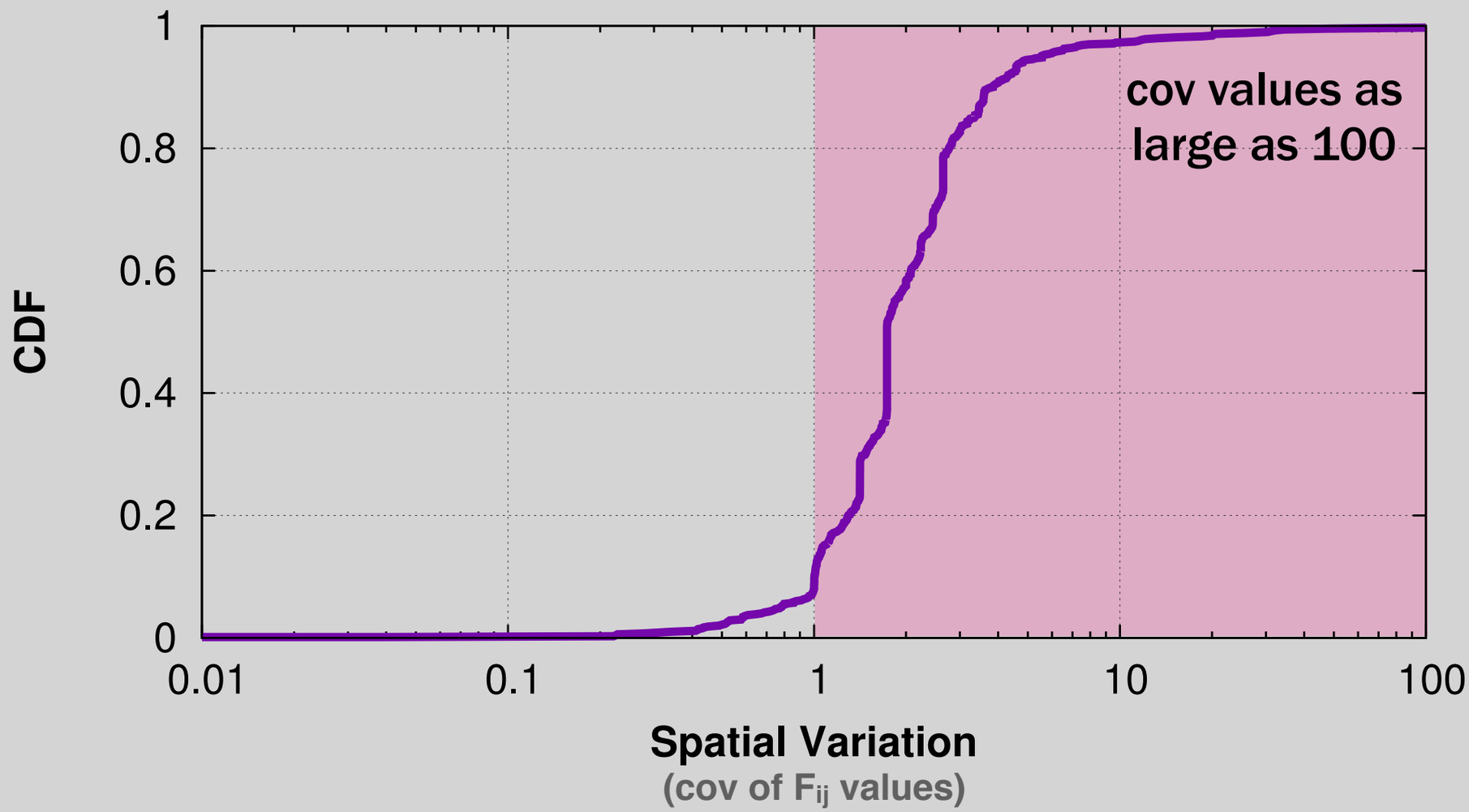
applications exhibit high spatial variability

SPATIAL VARIABILITY



applications exhibit high spatial variability

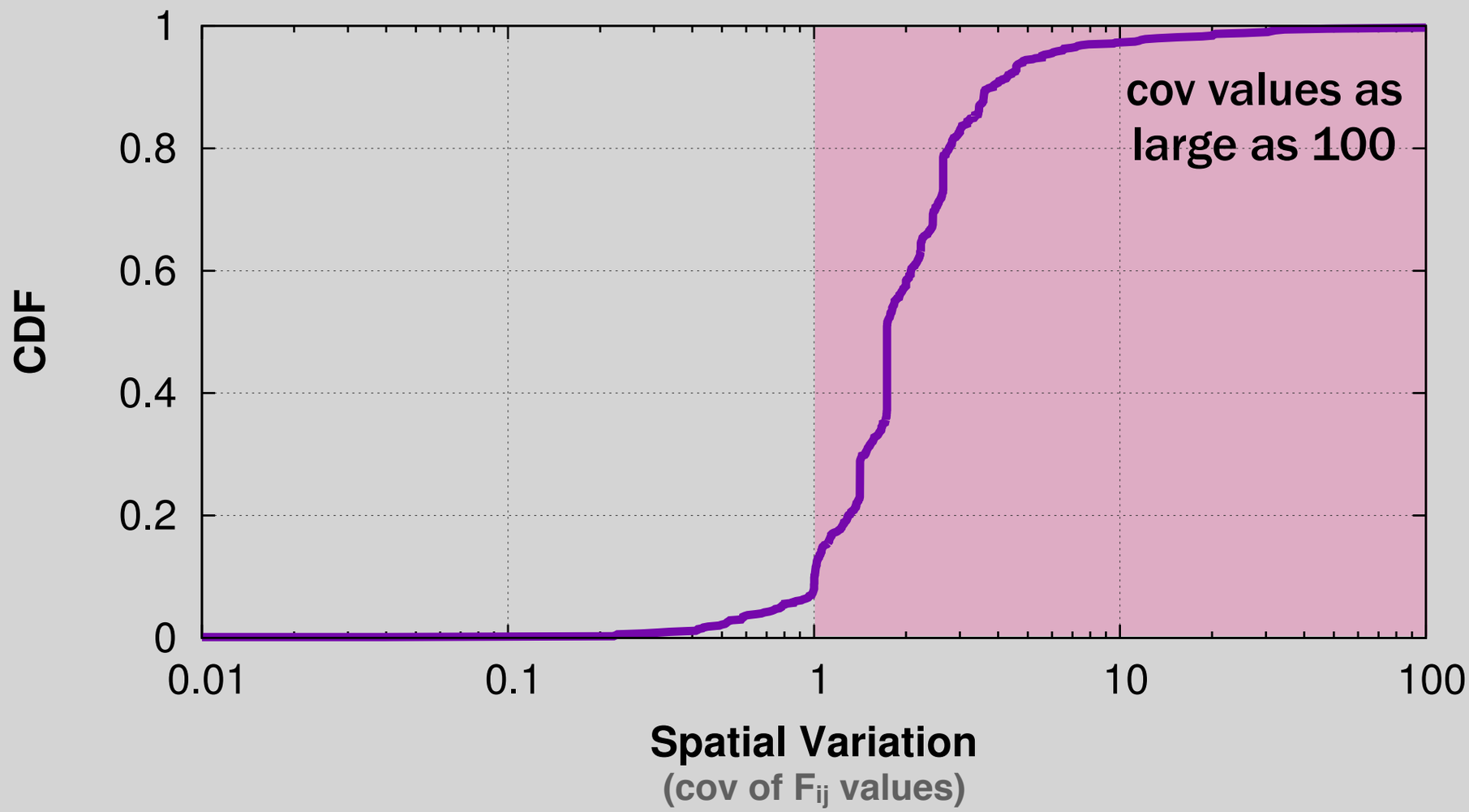
SPATIAL VARIABILITY



applications exhibit high spatial variability

(the same result holds for temporal variability, see paper)

SPATIAL VARIABILITY

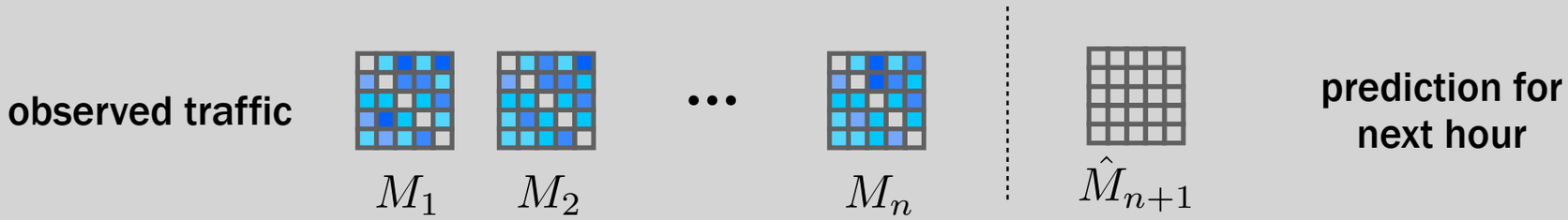


applications exhibit high spatial variability

(the same result holds for temporal variability, see paper)

customers cannot model such variability themselves, and existing models do not capture it

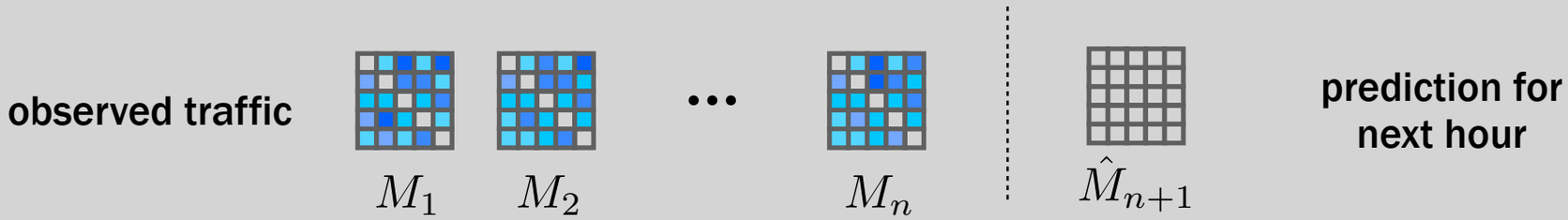
CICADA'S ALGORITHM



Cicada's prediction algorithm draws inspiration from the following works:

[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

CICADA'S ALGORITHM

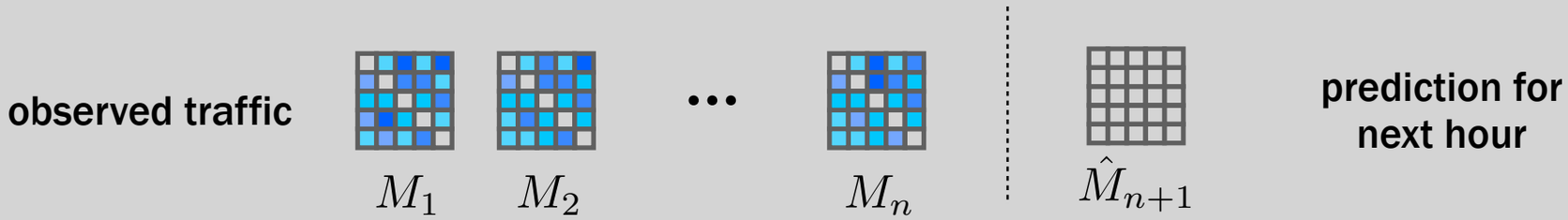


$$\hat{M}_{n+1} = f(M_1, M_2, \ldots, M_n)$$

Cicada's prediction algorithm draws inspiration from the following works:

[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

CICADA'S ALGORITHM

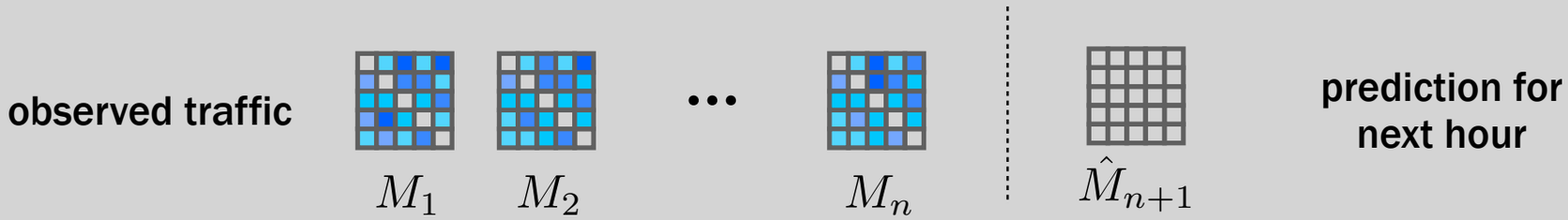


$$\hat{M}_{n+1} = w_1 \cdot M_1 + w_2 \cdot M_2 + \ldots + w_n \cdot M_n$$

Cicada's prediction algorithm draws inspiration from the following works:

[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

CICADA'S ALGORITHM

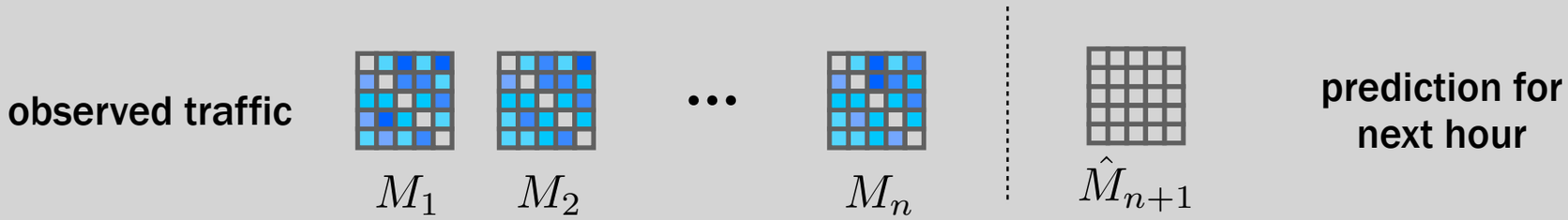


$$\hat{M}_{n+1} = w_1 \cdot M_1 + w_2 \cdot M_2 + \ldots + w_n \cdot M_n$$

instead of setting the weights beforehand, cicada learns the weights online; they are updated with every prediction based on errors made in previous predictions

Cicada's prediction algorithm draws inspiration from the following works:
[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

CICADA'S ALGORITHM

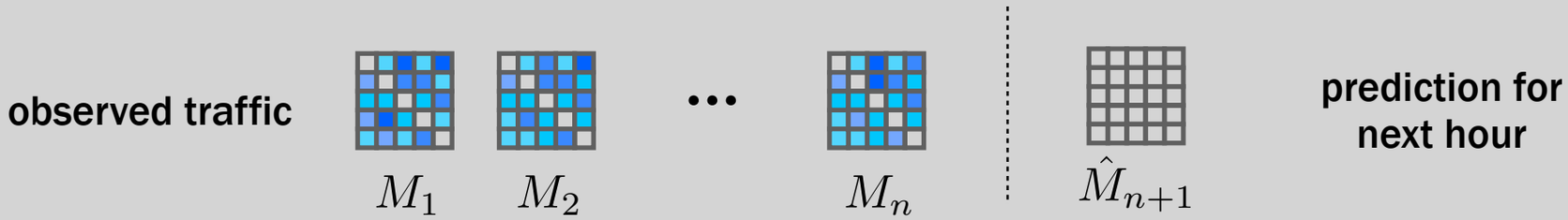


$$\hat{M}_{n+1} = w_1(n) \cdot M_1 + w_2(n) \cdot M_2 + \ldots + w_n(n) \cdot M_n$$

instead of setting the weights beforehand, cicada learns the weights online; they are updated with every prediction based on errors made in previous predictions

Cicada's prediction algorithm draws inspiration from the following works:
[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

CICADA'S ALGORITHM



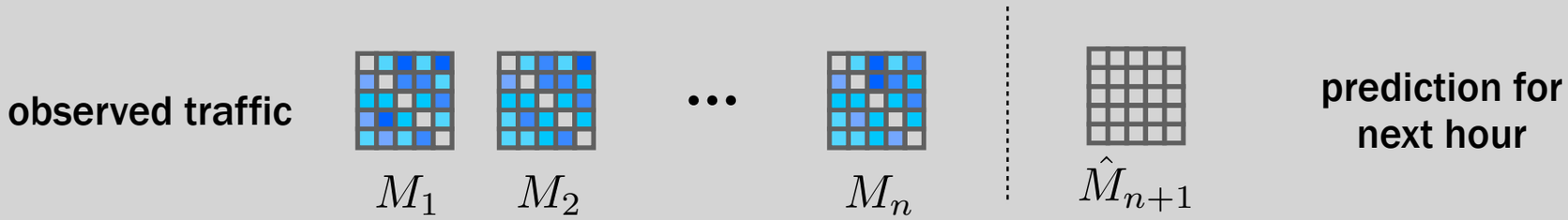
$$\hat{M}_{n+1} = w_1(n) \cdot M_1 + w_2(n) \cdot M_2 + \ldots + w_n(n) \cdot M_n$$

instead of setting the weights beforehand, cicada learns the weights online; they are updated with every prediction based on errors made in previous predictions

$$w_i(n+1) = w_i(n)$$

Cicada's prediction algorithm draws inspiration from the following works:
[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

CICADA'S ALGORITHM



$$\hat{M}_{n+1} = w_1(n) \cdot M_1 + w_2(n) \cdot M_2 + \ldots + w_n(n) \cdot M_n$$

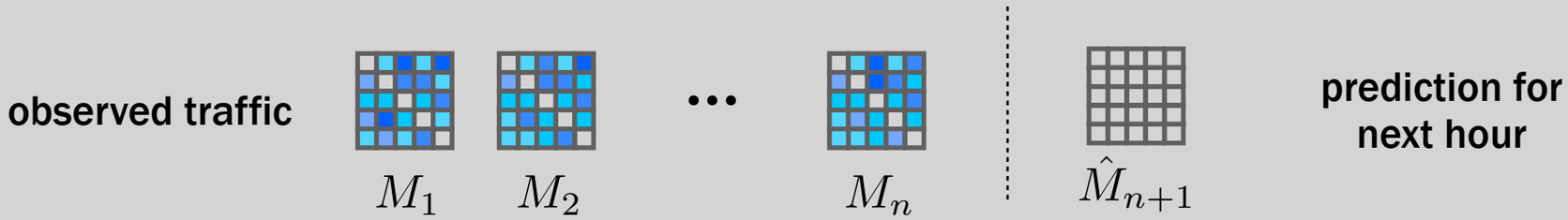
instead of setting the weights beforehand, cicada learns the weights online; they are updated with every prediction based on errors made in previous predictions

loss function

$$w_i(n+1) = w_i(n) \cdot e^{-L(i,n)}$$

Cicada's prediction algorithm draws inspiration from the following works:
[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

CICADA'S ALGORITHM



$$\hat{M}_{n+1} = w_1(n) \cdot M_1 + w_2(n) \cdot M_2 + \ldots + w_n(n) \cdot M_n$$

instead of setting the weights beforehand, cicada learns the weights online; they are updated with every prediction based on errors made in previous predictions

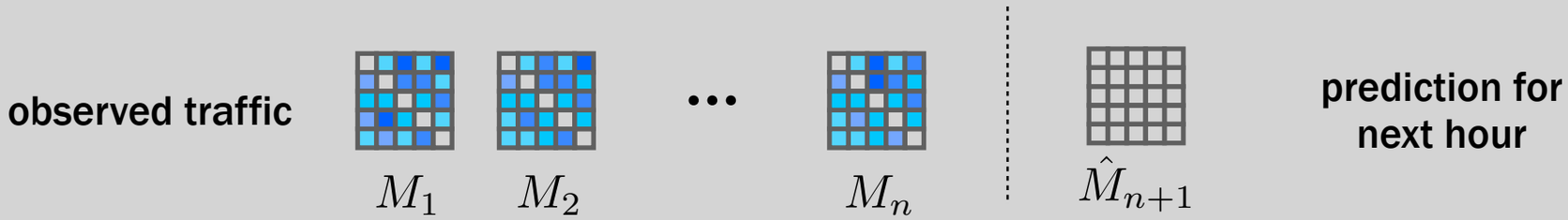
loss function

$$w_i(n+1) = w_i(n) \cdot e^{-L(i,n)}$$

$$L(i,n) = \frac{\|\vec{M}_i - \vec{M}_n\|}{\|\vec{M}_n\|}$$

Cicada's prediction algorithm draws inspiration from the following works:
[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

CICADA'S ALGORITHM



$$\hat{M}_{n+1} = w_1(n) \cdot M_1 + w_2(n) \cdot M_2 + \ldots + w_n(n) \cdot M_n$$

instead of setting the weights beforehand, cicada learns the weights online; they are updated with every prediction based on errors made in previous predictions

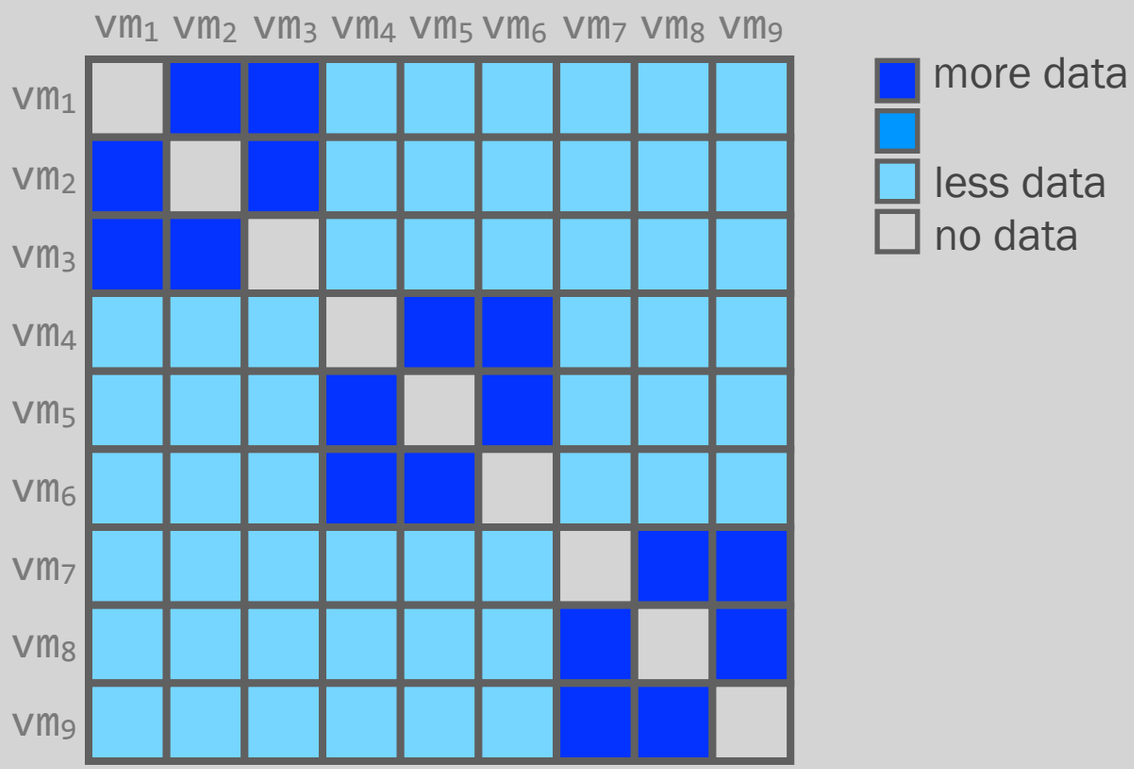
$$w_i(n+1) = w_i(n) \cdot \underbrace{e^{-L(i,n)}}_{\text{loss function}} \cdot \underbrace{\frac{1}{Z_{n+1}}}_{\text{normalization}} \qquad L(i,n) = \frac{\|\vec{M}_i - \vec{M}_n\|}{\|\vec{M}_n\|}$$

Cicada's prediction algorithm draws inspiration from the following works:

[1] Wang, et al. "COPE: Traffic Engineering in Dynamic Networks". SIGCOMM 2006.
[2] Herbster and Warmuth. "Tracking the Best Expert". Machine Learning Vol 32, 1998.

EVALUATION

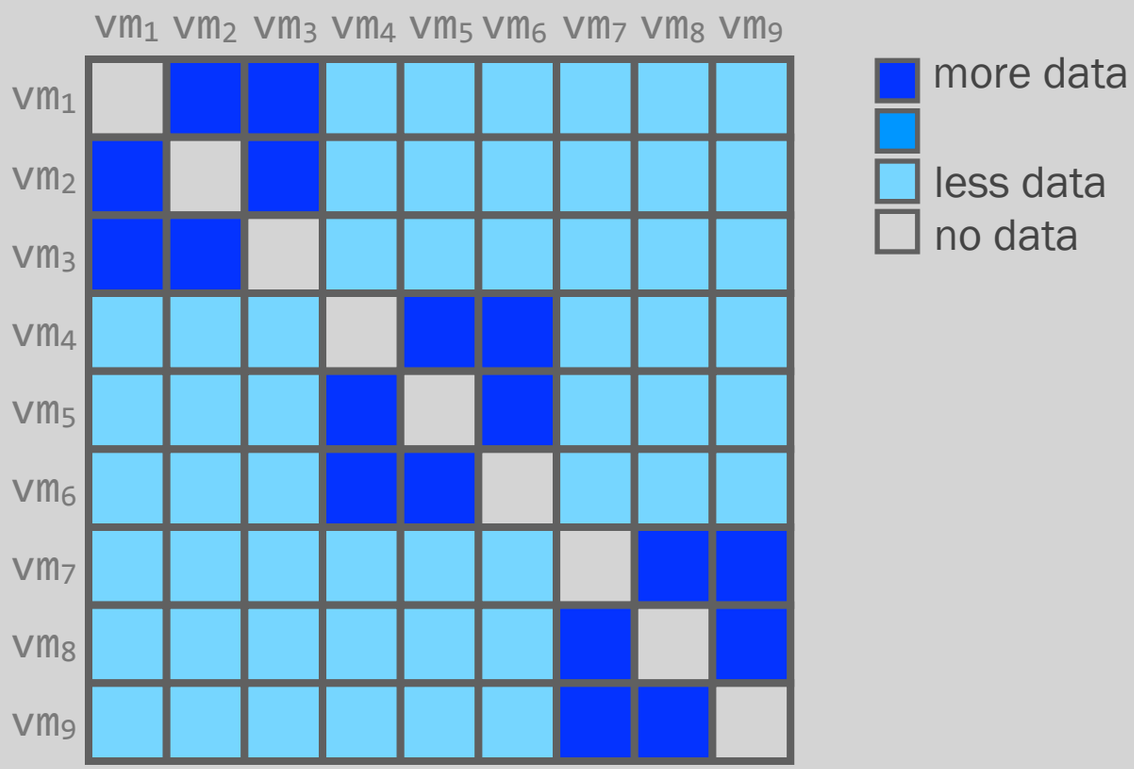
to evaluate cicada, we compared its predictions to ones made by a VOC-style system [1]



[1] Ballani, et al. “Towards Predictable Datacenter Networks”. SIGCOMM 2011.

EVALUATION

to evaluate cicada, we compared its predictions to ones made by a VOC-style system [1]



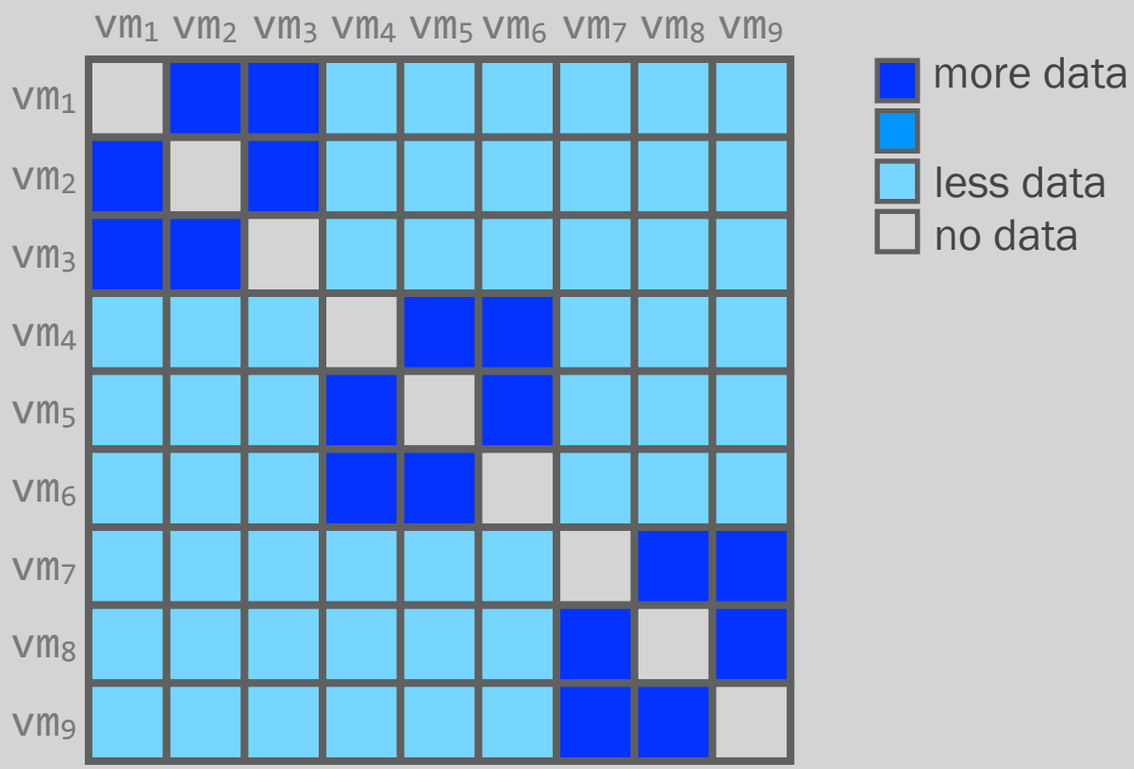
VOC requires customers to input parameters

(group sizes, amount of bandwidth)

[1] Ballani, et al. “Towards Predictable Datacenter Networks”. SIGCOMM 2011.

EVALUATION

to evaluate cicada, we compared its predictions to ones made by a VOC-style system [1]



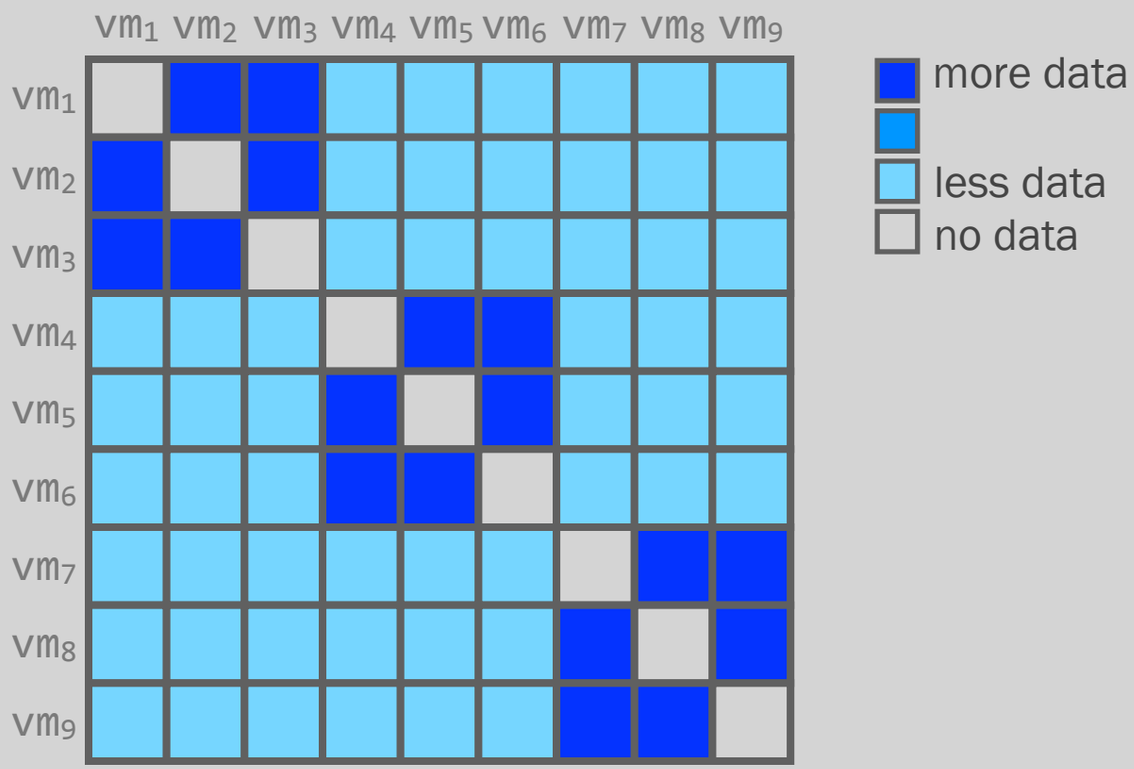
VOC requires customers to input parameters
(group sizes, amount of bandwidth)

we used an oracle method to pick the best possible parameters, and also allowed for temporal variability

[1] Ballani, et al. “Towards Predictable Datacenter Networks”. SIGCOMM 2011.

EVALUATION

to evaluate cicada, we compared its predictions to ones made by a VOC-style system [1]



VOC requires customers to input parameters
(group sizes, amount of bandwidth)

we used an oracle method to pick the best possible parameters, and also allowed for temporal variability

---

L2-norm error

$$E = \frac{\|\hat{M} - M\|}{\|\hat{M}\|}$$

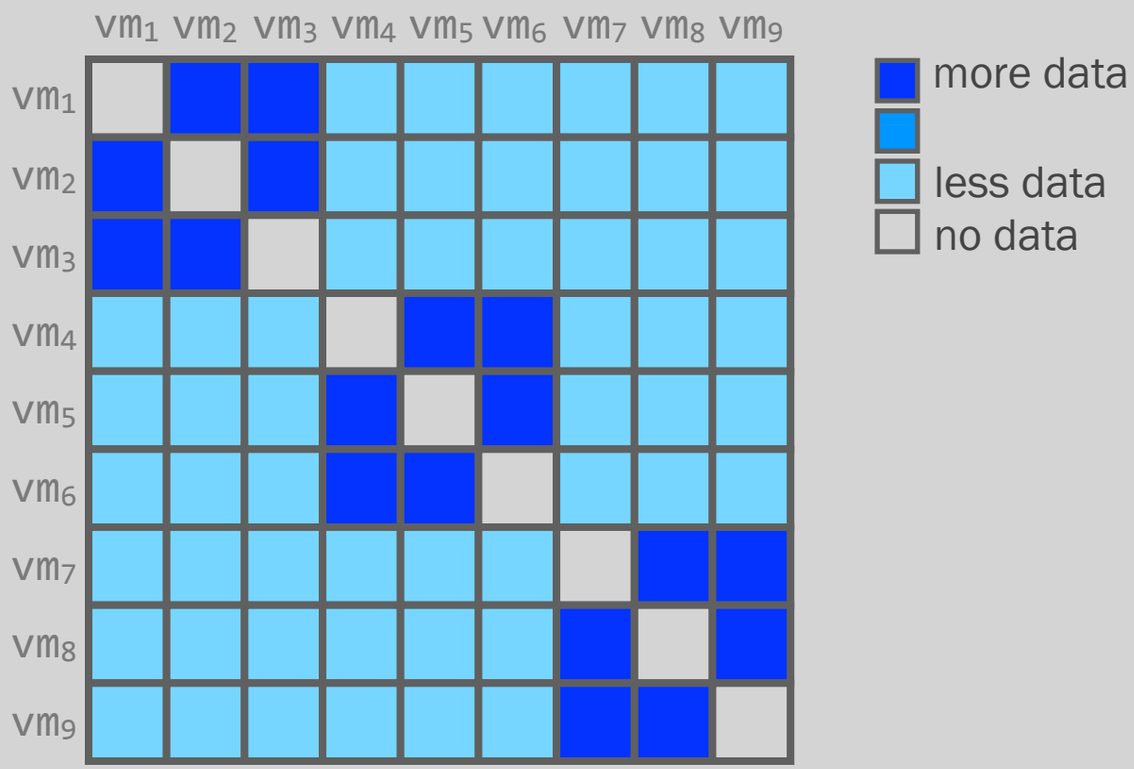
relative error

$$E = \frac{\sum_{i=1}^{i=n^2} \hat{M}[i] - M[i]}{\sum_{i=1}^{i=n^2} M[i]}$$

[1] Ballani, et al. “Towards Predictable Datacenter Networks”. SIGCOMM 2011.

EVALUATION

to evaluate cicada, we compared its predictions to ones made by a VOC-style system [1]



VOC requires customers to input parameters
(group sizes, amount of bandwidth)

we used an oracle method to pick the best possible parameters, and also allowed for temporal variability

---

L2-norm error

$$E = \frac{\|\hat{M} - M\|}{\|\hat{M}\|}$$

relative error

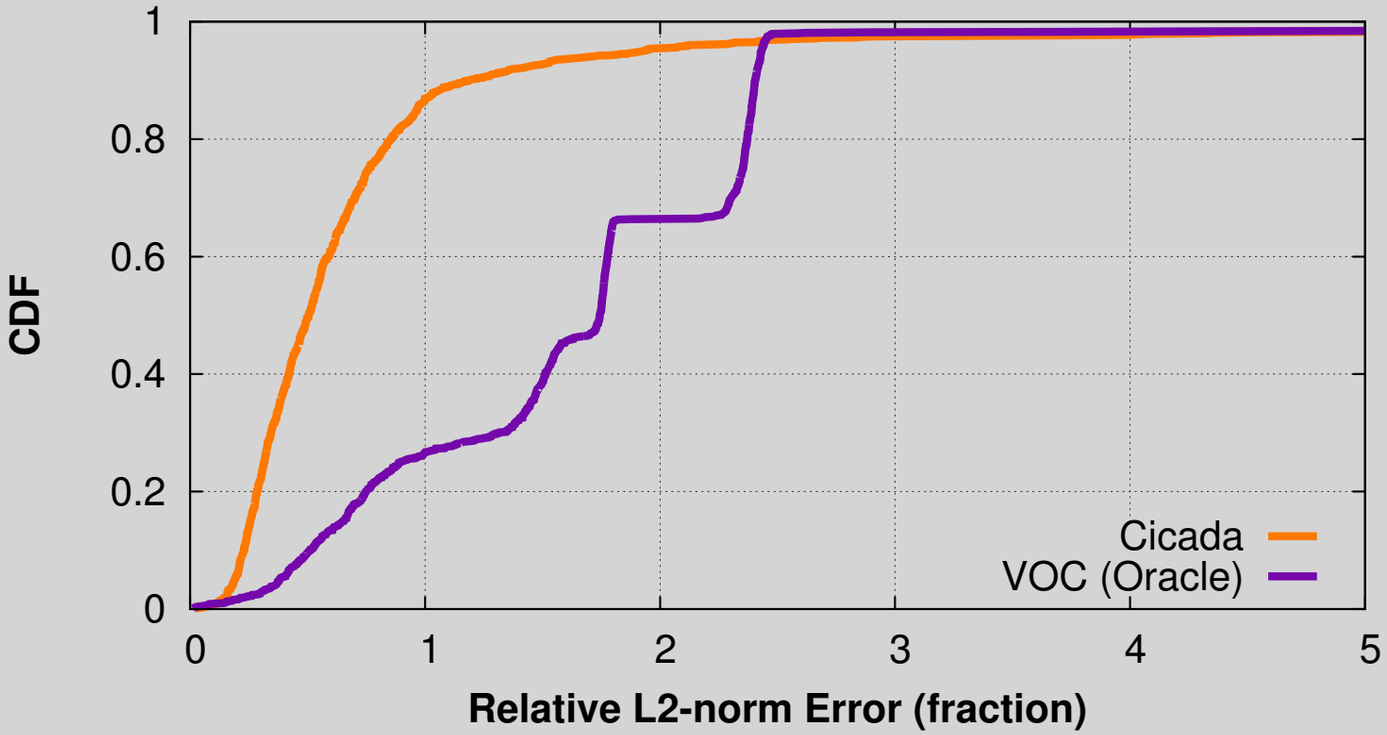
$$E = \frac{\sum_{i=1}^{i=n^2} \hat{M}[i] - M[i]}{\sum_{i=1}^{i=n^2} M[i]}$$

← **differentiates between over- and under-prediction**

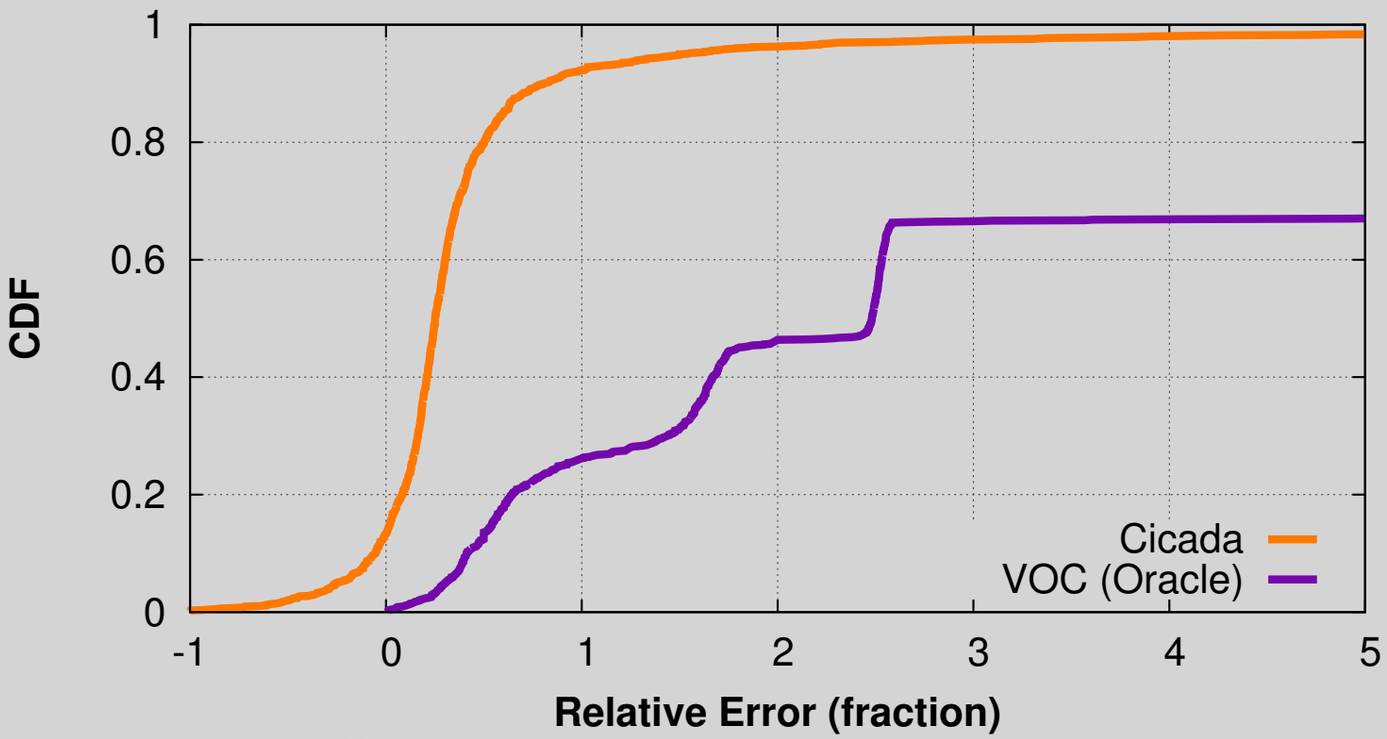
[1] Ballani, et al. “Towards Predictable Datacenter Networks”. SIGCOMM 2011.

RESULTS

PREDICTING AVERAGE DEMAND



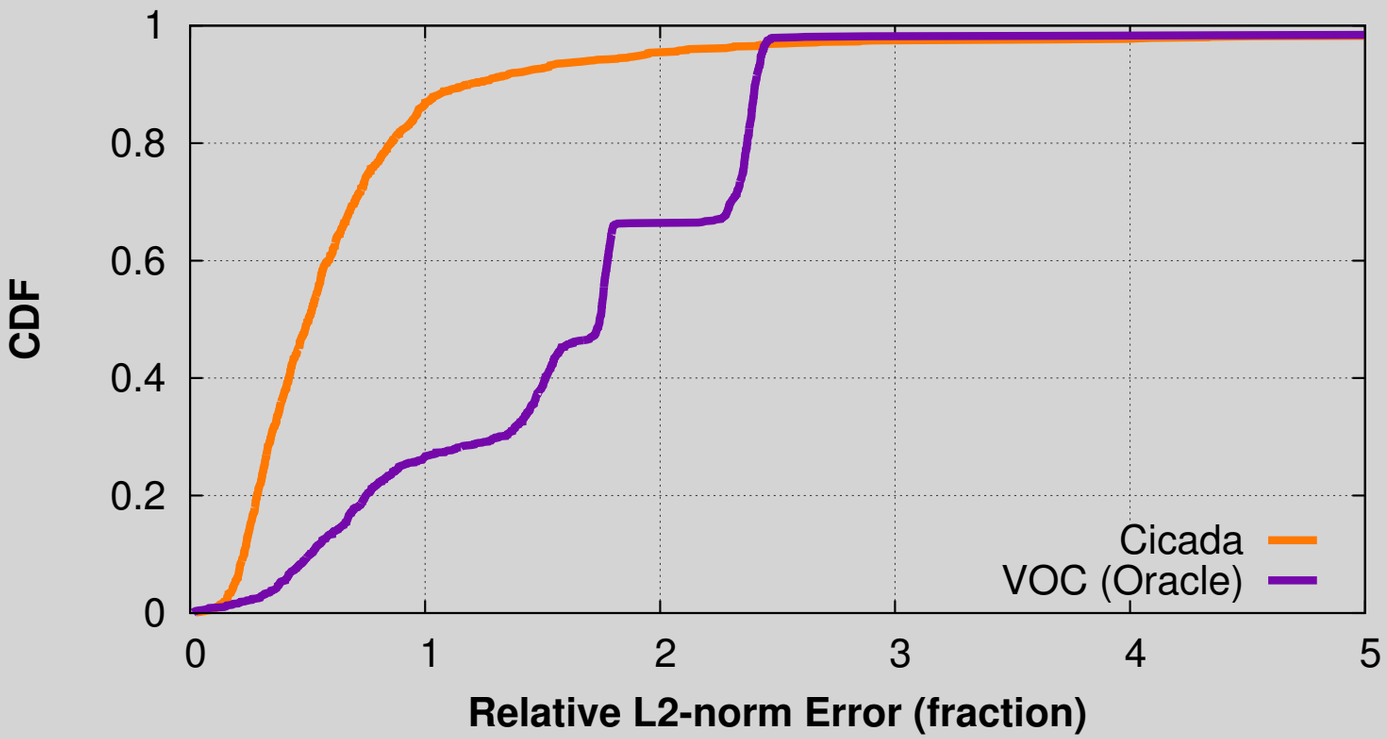
does not differentiate between over- and under-prediction



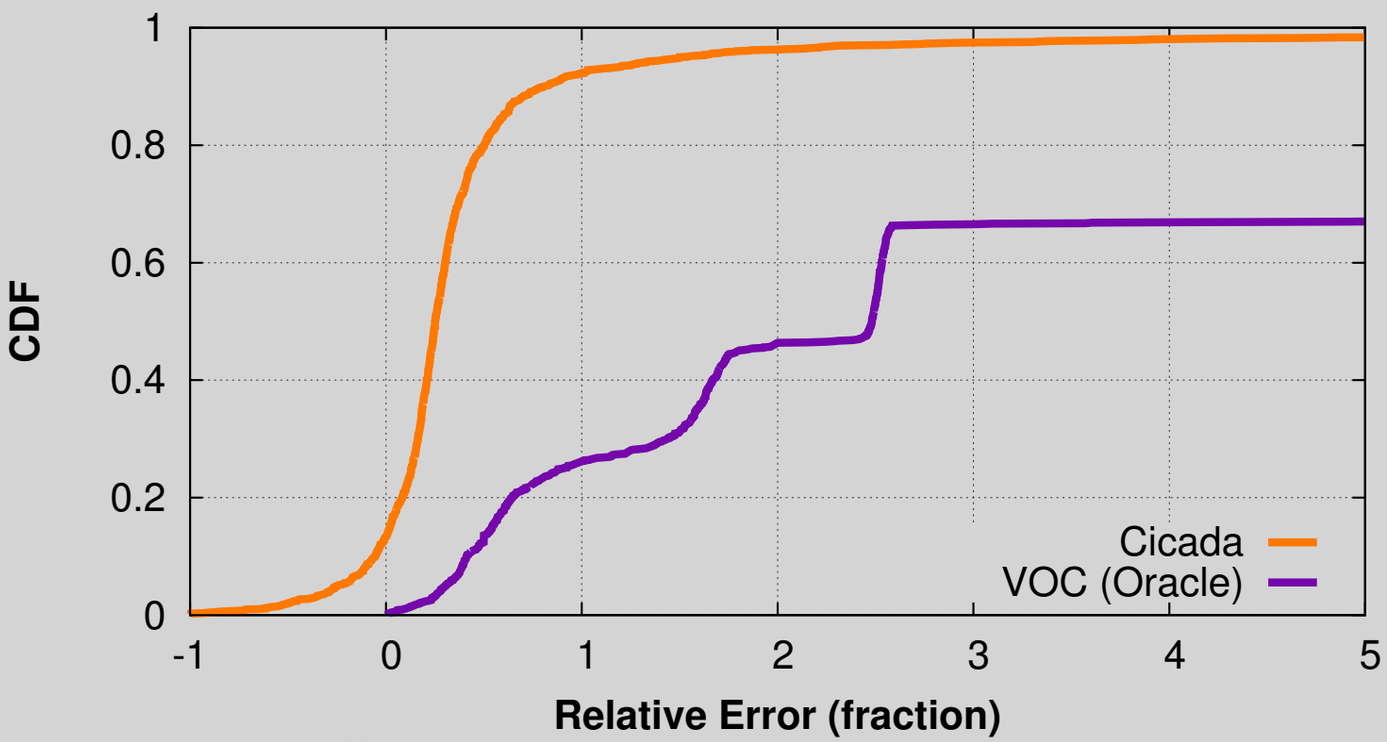
differentiates between over- and under-prediction

RESULTS

PREDICTING AVERAGE DEMAND



does not differentiate between over- and under-prediction



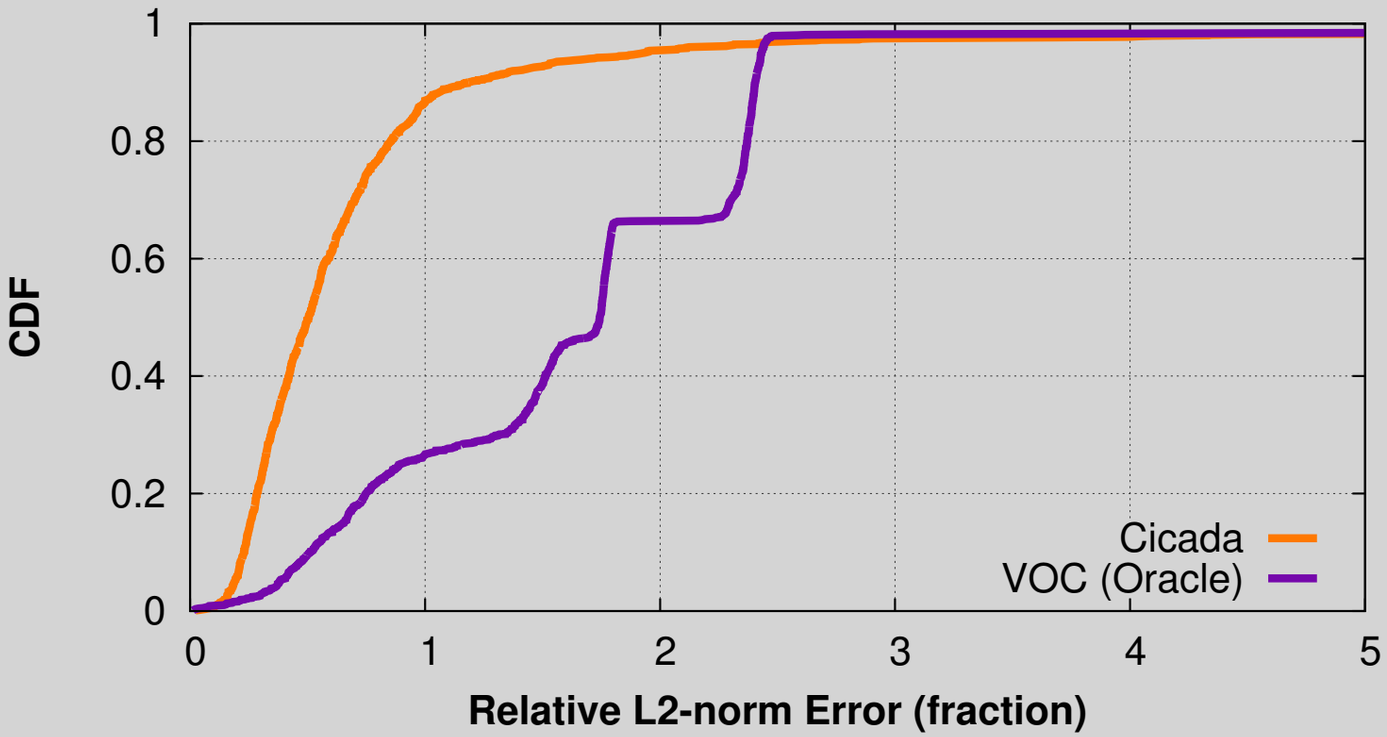
differentiates between over- and under-prediction

cicada's predictions outperform VOC-style predictions (median error decreases by 90%) **and require no customer input**

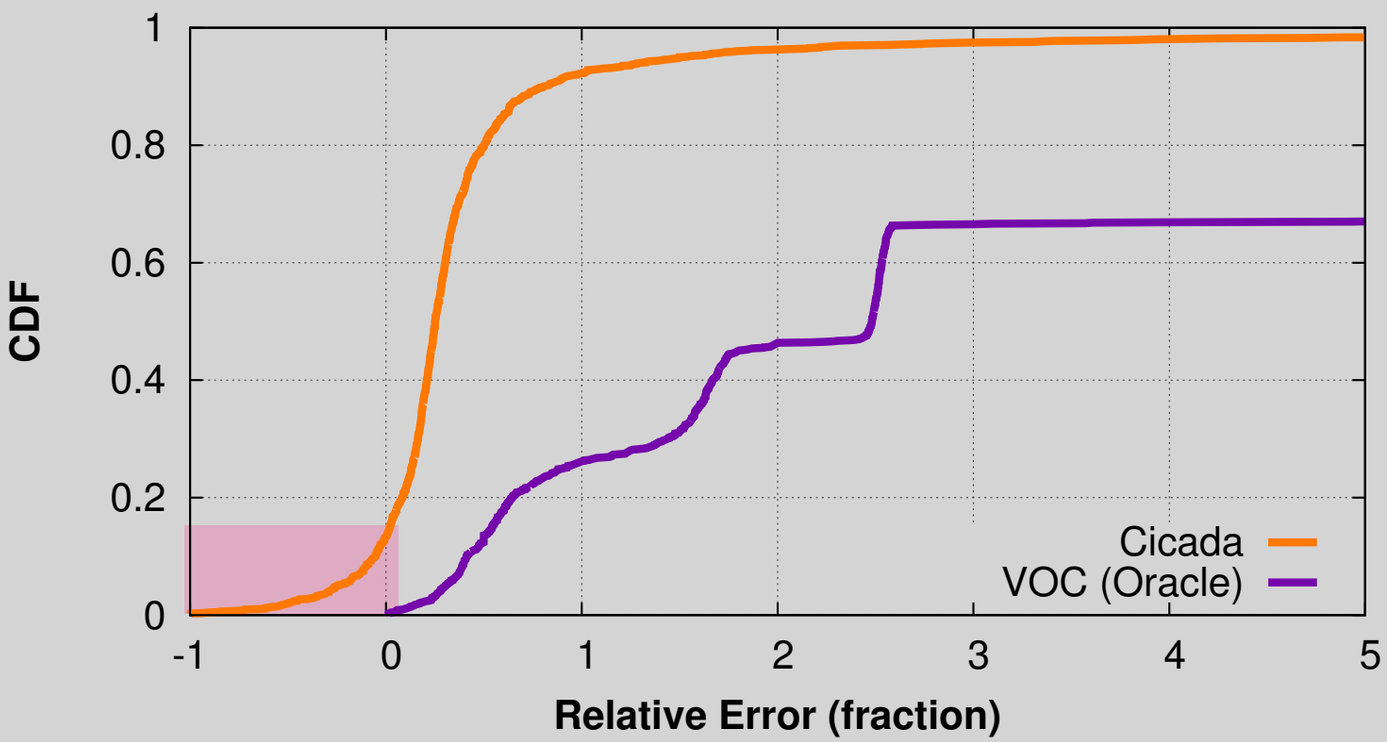
(71% for L2 error)

RESULTS

PREDICTING AVERAGE DEMAND



does not differentiate between over- and under-prediction



differentiates between over- and under-prediction

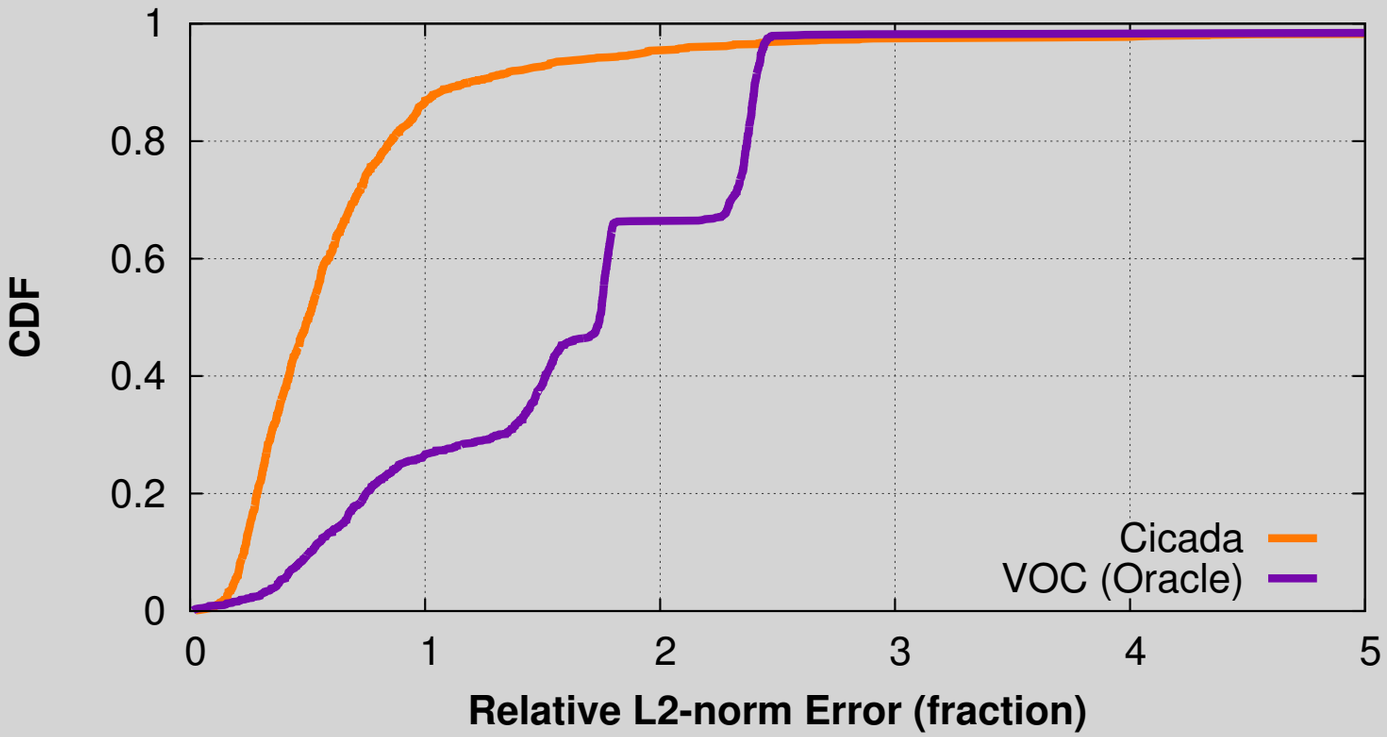
cicada's predictions outperform VOC-style predictions (median error decreases by 90%) **and require no customer input**

(71% for L2 error)

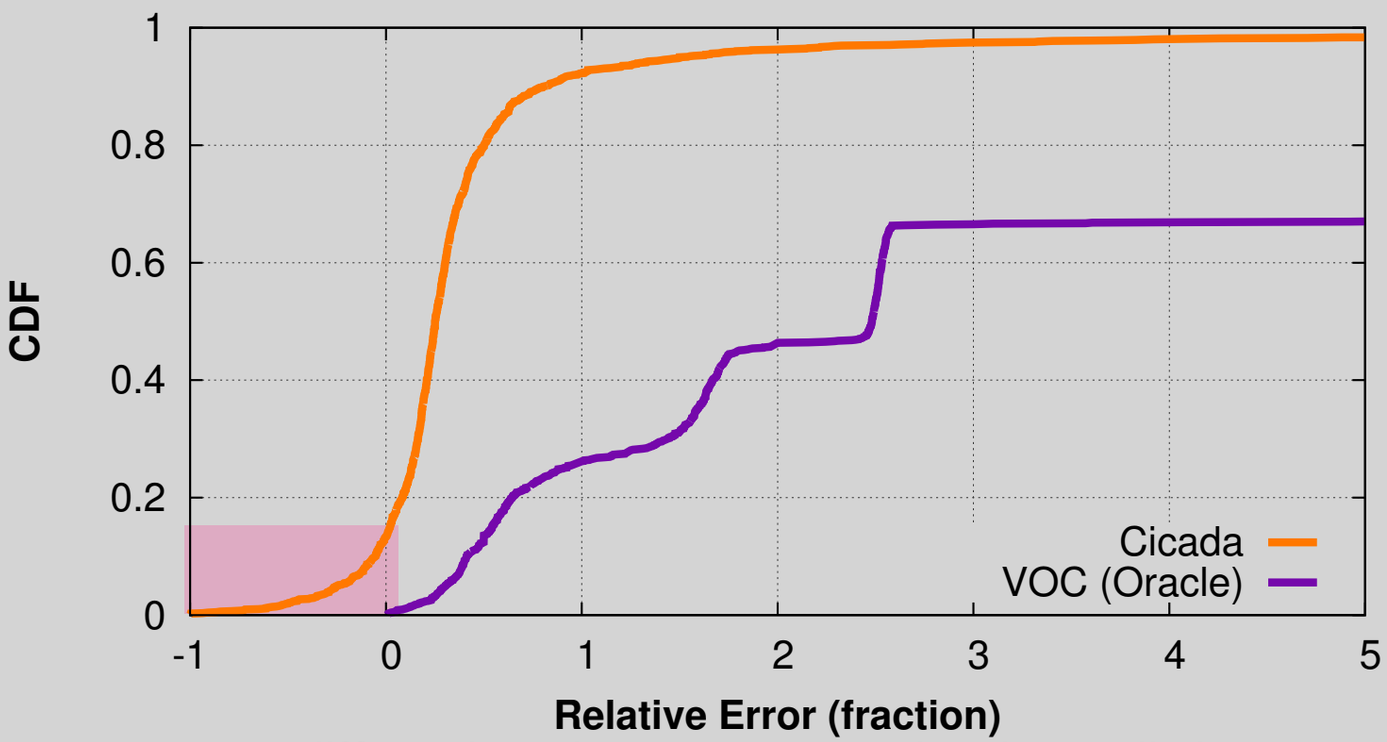
cicada occasionally under-predicts; VOC does not because we selected its parameters to avoid under-prediction

RESULTS

PREDICTING AVERAGE DEMAND



does not differentiate between over- and under-prediction



differentiates between over- and under-prediction

cicada's predictions outperform VOC-style predictions (median error decreases by 90%) **and require no customer input**

(71% for L2 error)

cicada occasionally under-predicts; VOC does not because we selected its parameters to avoid under-prediction

cicada's predictions typically require only 1-2 hours of application history

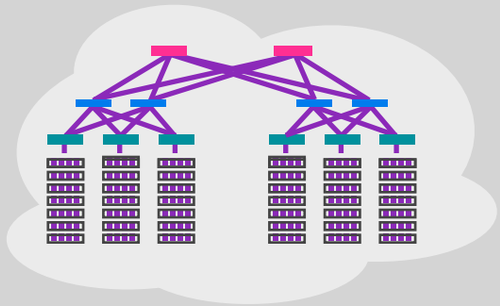
PREDICTIONS → GUARANTEES

cicada turns its predictions into pipe-model guarantees;
a separate guarantee for each (source, destination) pair

PREDICTIONS → GUARANTEES

cicada turns its predictions into pipe-model guarantees;
a separate guarantee for each (source, destination) pair

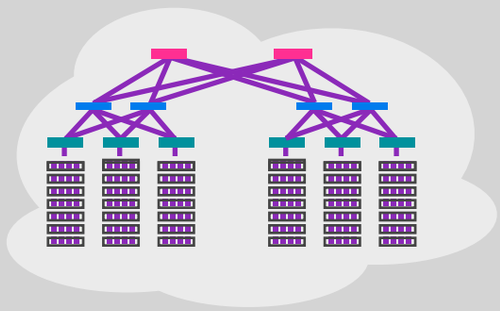
network resources must be available



PREDICTIONS → GUARANTEES

cicada turns its predictions into pipe-model guarantees;
a separate guarantee for each (source, destination) pair

network resources must be available

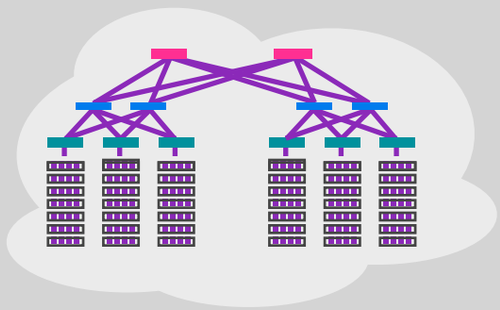


customers can add a buffer to the offered guarantees

PREDICTIONS → GUARANTEES

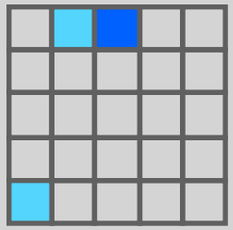
cicada turns its predictions into pipe-model guarantees;
a separate guarantee for each (source, destination) pair

network resources must be available



customers can add a buffer to the offered guarantees

the provider may choose to augment predictions that cicada is not confident about

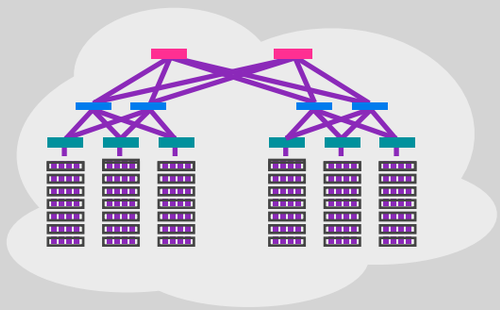


typically not confident for "small" applications

PREDICTIONS → GUARANTEES

cicada turns its predictions into pipe-model guarantees;
a separate guarantee for each (source, destination) pair

network resources must be available

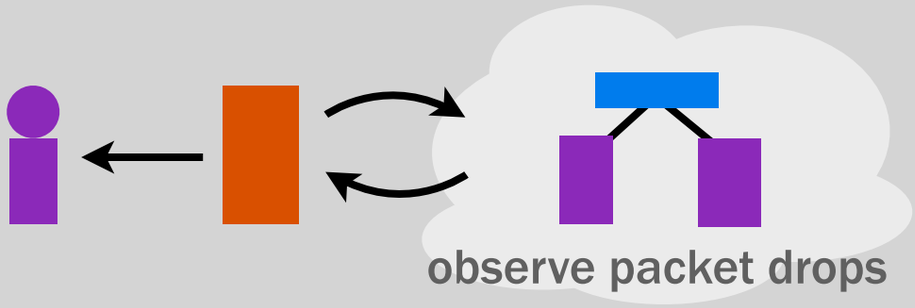


customers can add a buffer to the offered guarantees

the provider may choose to augment predictions that cicada is not confident about

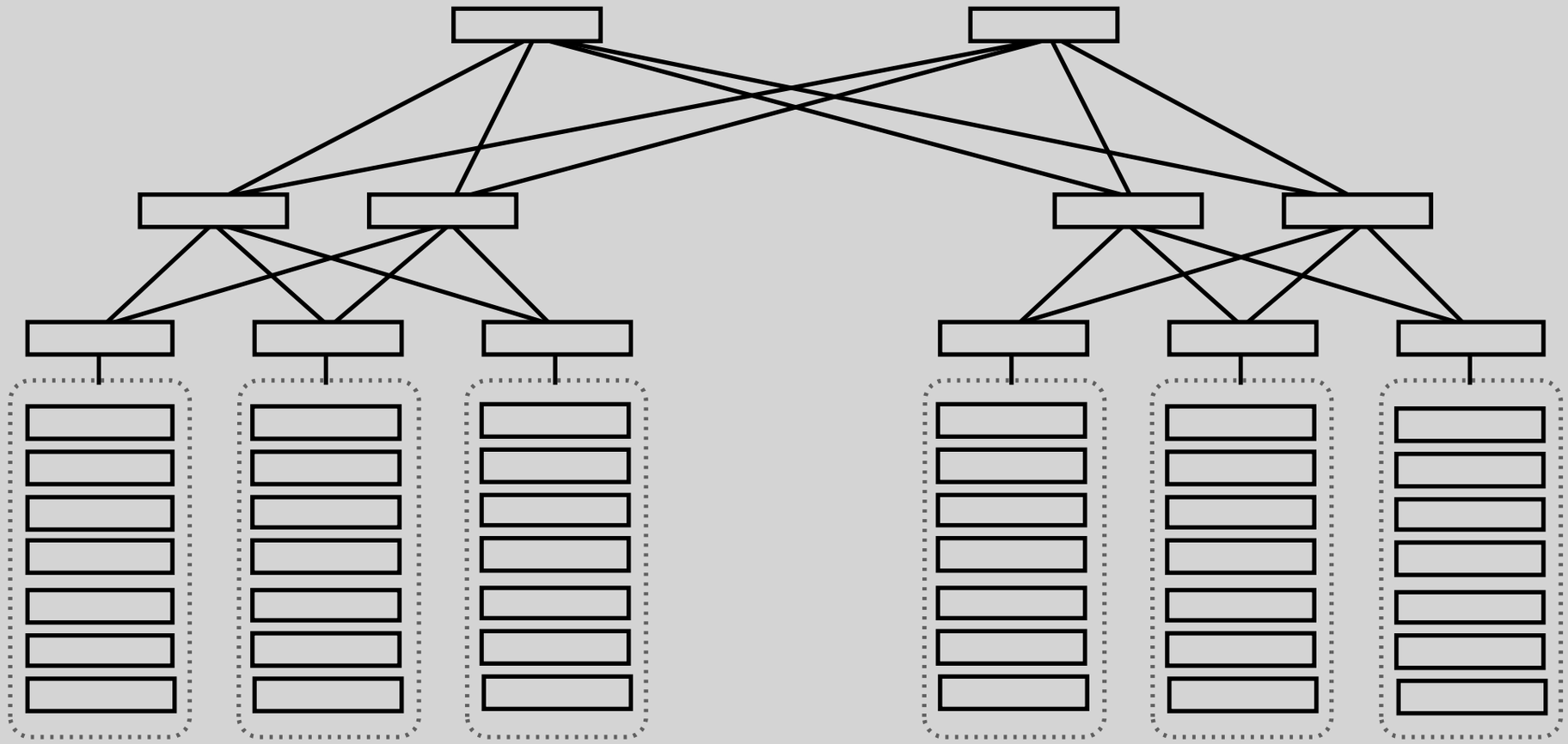
typically not confident for "small" applications

cicada can detect when a guarantee is too low, and offer a new one



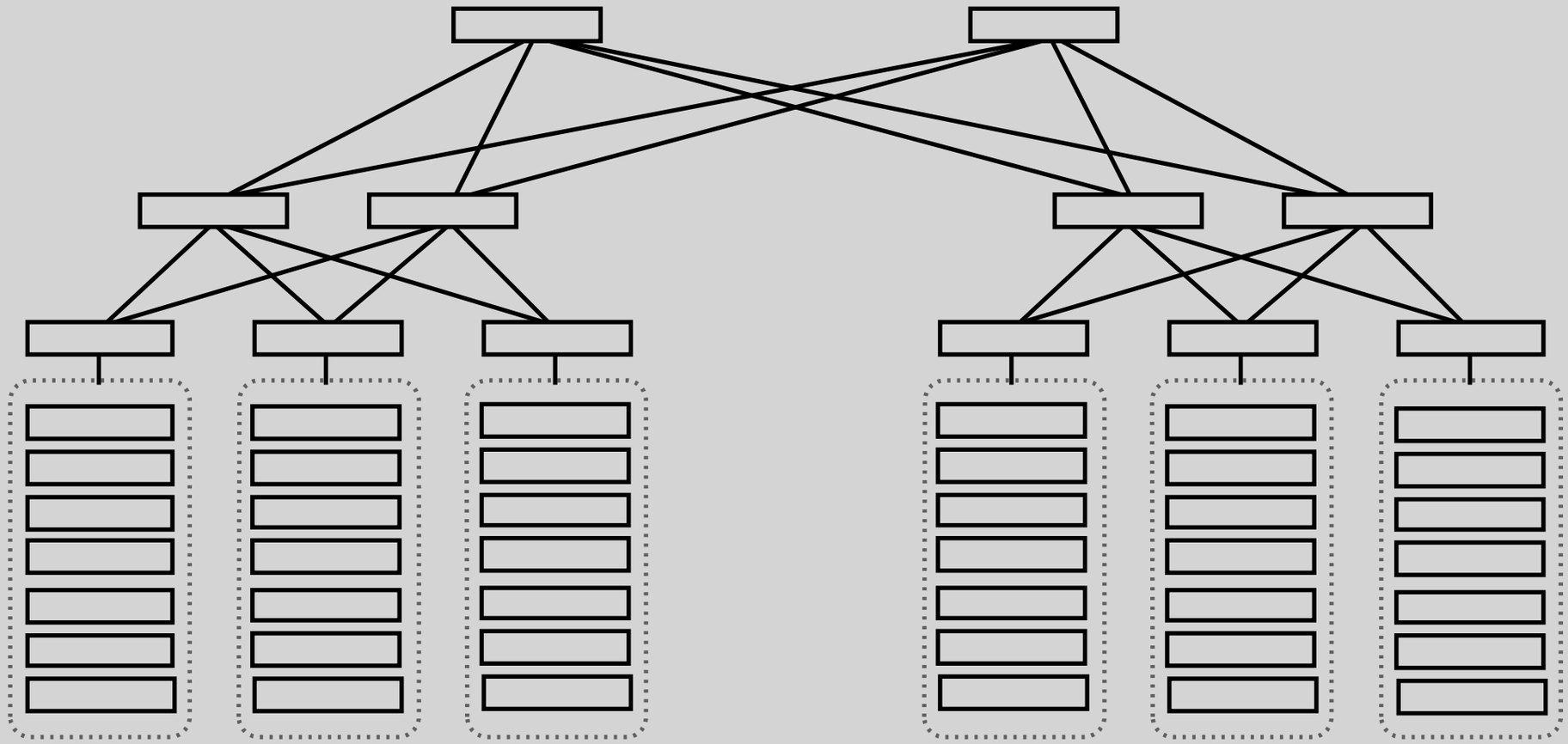
NETWORK UTILIZATION

do cicada's bandwidth guarantees improve network utilization?



NETWORK UTILIZATION

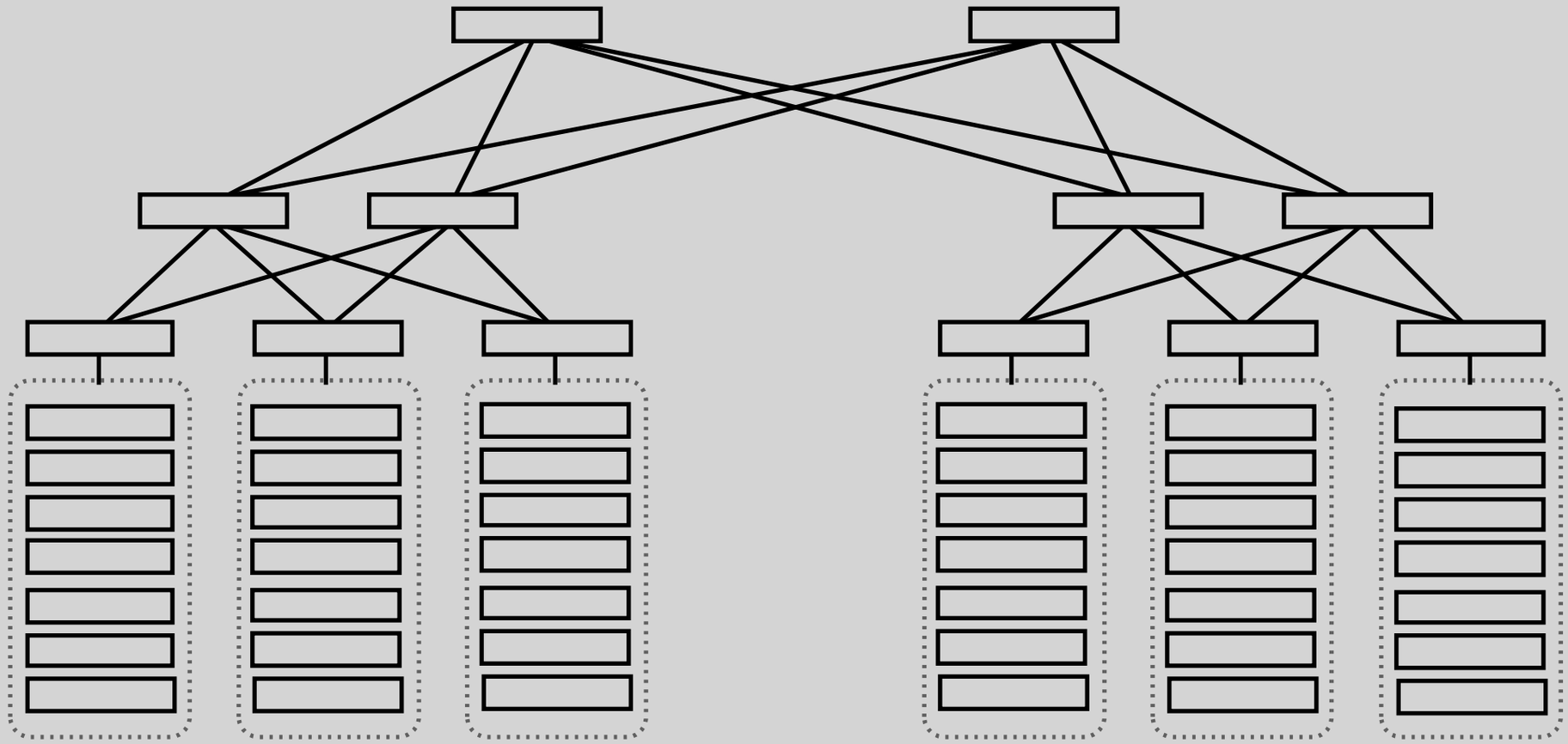
do cicada's bandwidth guarantees improve network utilization?



wasted bandwidth: bandwidth that is guaranteed for an application, but not *used* by the application

NETWORK UTILIZATION

do cicada's bandwidth guarantees improve network utilization?



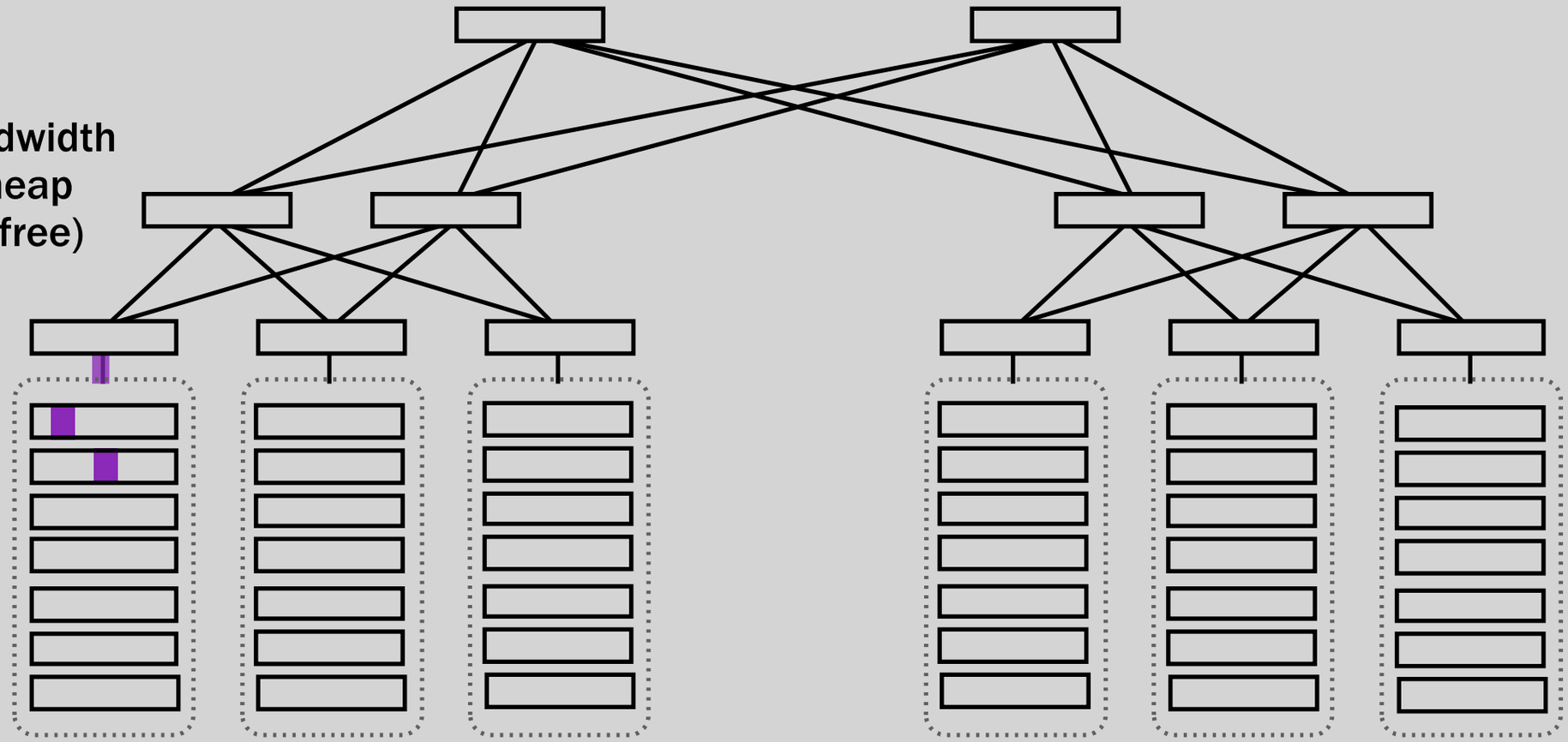
wasted bandwidth: bandwidth that is guaranteed for an application, but not *used* by the application

provider's goal: minimize wasted bandwidth

NETWORK UTILIZATION

do cicada's bandwidth guarantees improve network utilization?

intra-rack bandwidth
is usually cheap
(sometimes free)

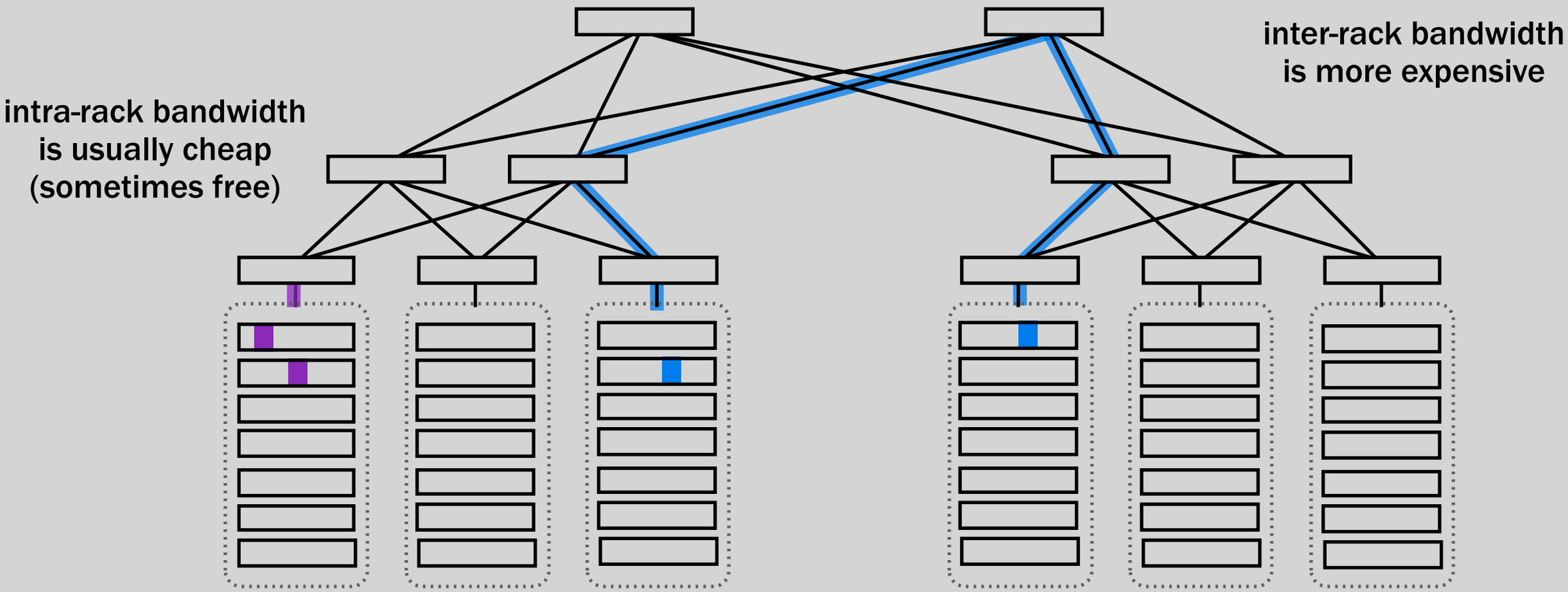


wasted bandwidth: bandwidth that is guaranteed
for an application, but not *used* by the application

provider's goal: minimize wasted bandwidth

NETWORK UTILIZATION

do cicada's bandwidth guarantees improve network utilization?

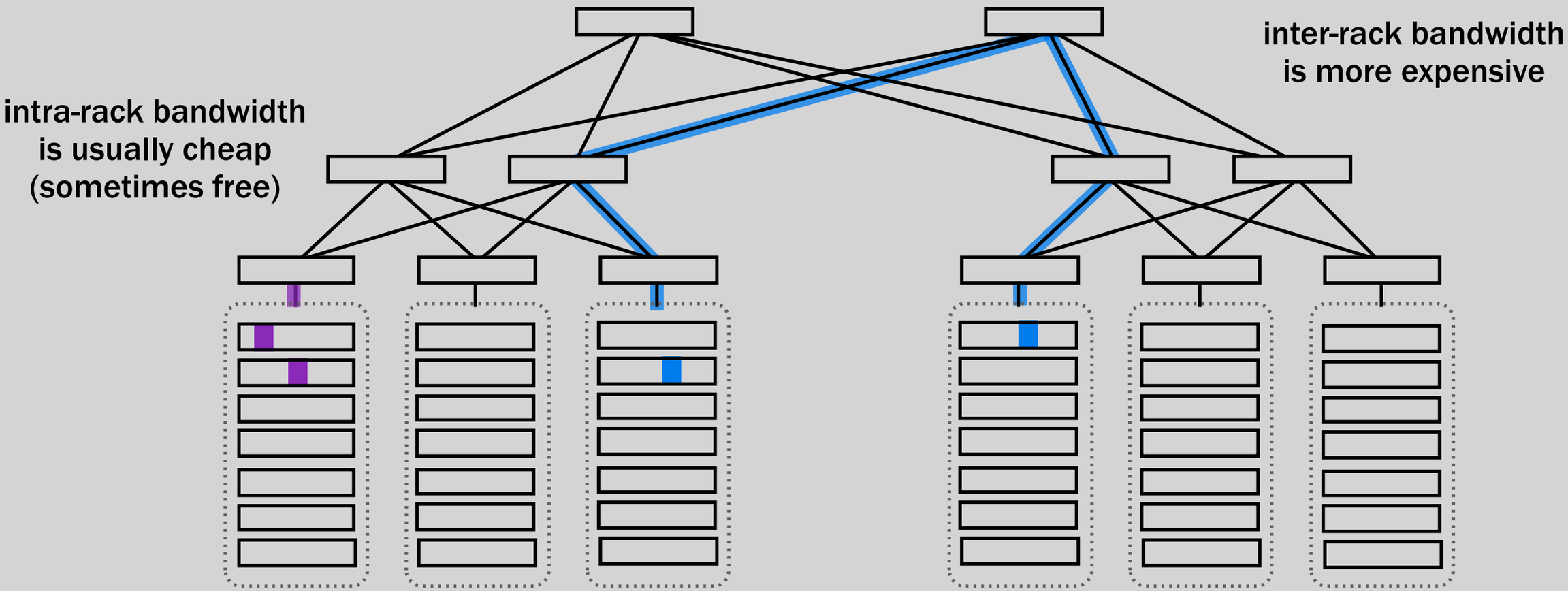


wasted bandwidth: bandwidth that is guaranteed for an application, but not *used* by the application

provider's goal: minimize wasted bandwidth

NETWORK UTILIZATION

do cicada's bandwidth guarantees improve network utilization?

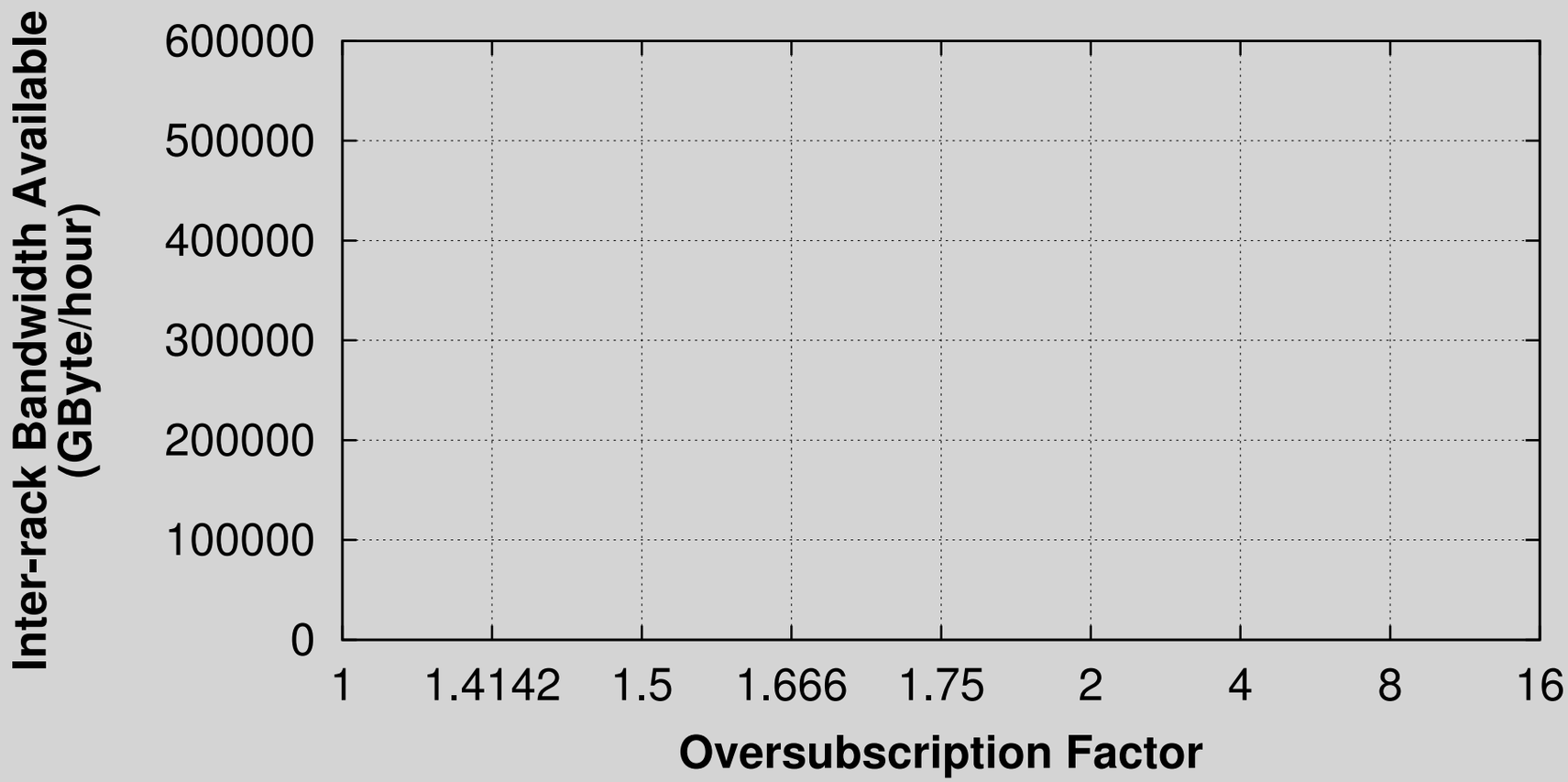


wasted bandwidth: bandwidth that is guaranteed for an application, but not *used* by the application

provider's goal: minimize wasted *inter-rack* bandwidth

NETWORK UTILIZATION

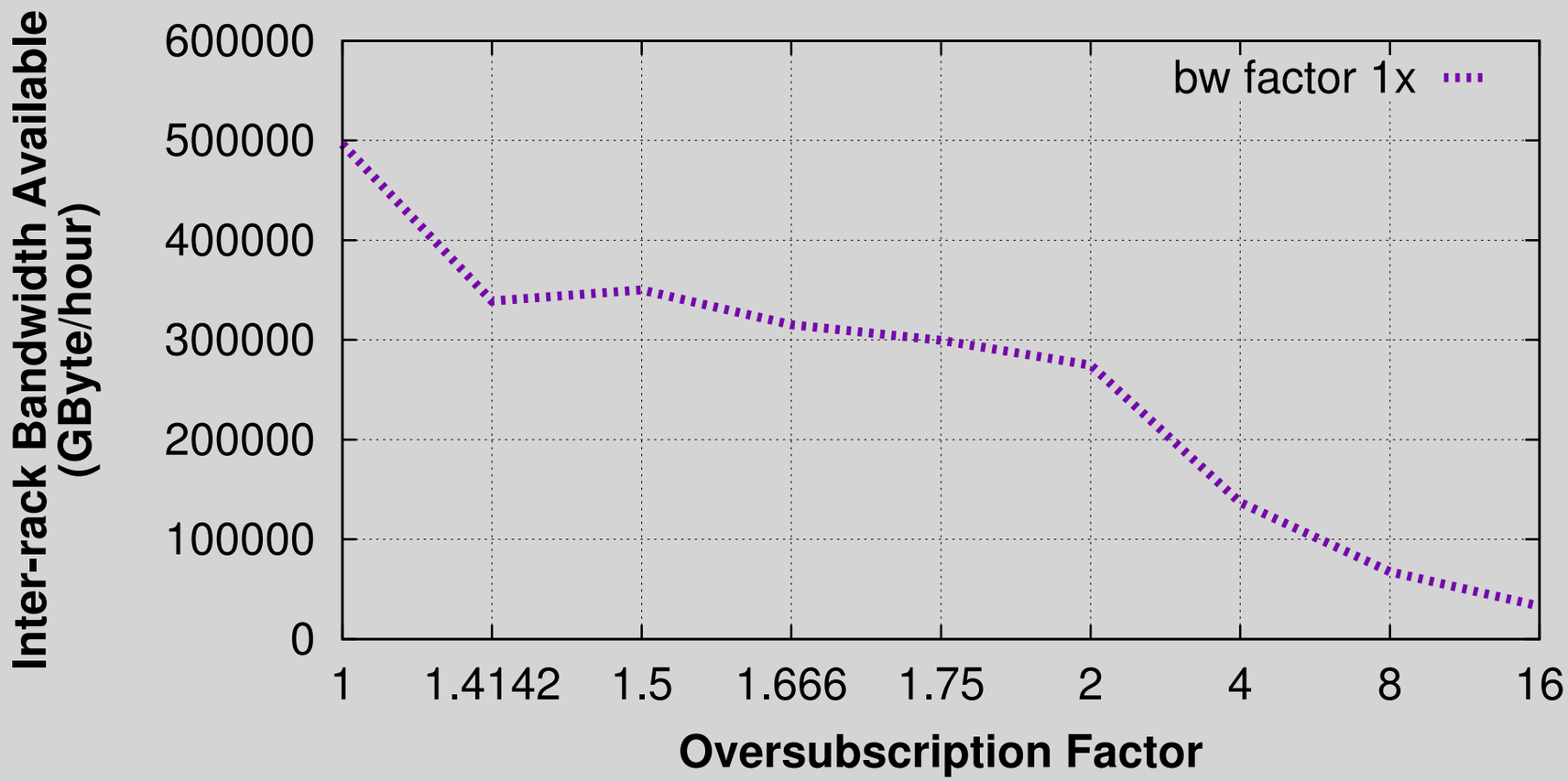
cicada's greedy placement heuristic: place the pairs of VMs that need the highest guarantees on the fastest paths (see paper for details)



[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

NETWORK UTILIZATION

cicada's greedy placement heuristic: place the pairs of VMs that need the highest guarantees on the fastest paths (see paper for details)

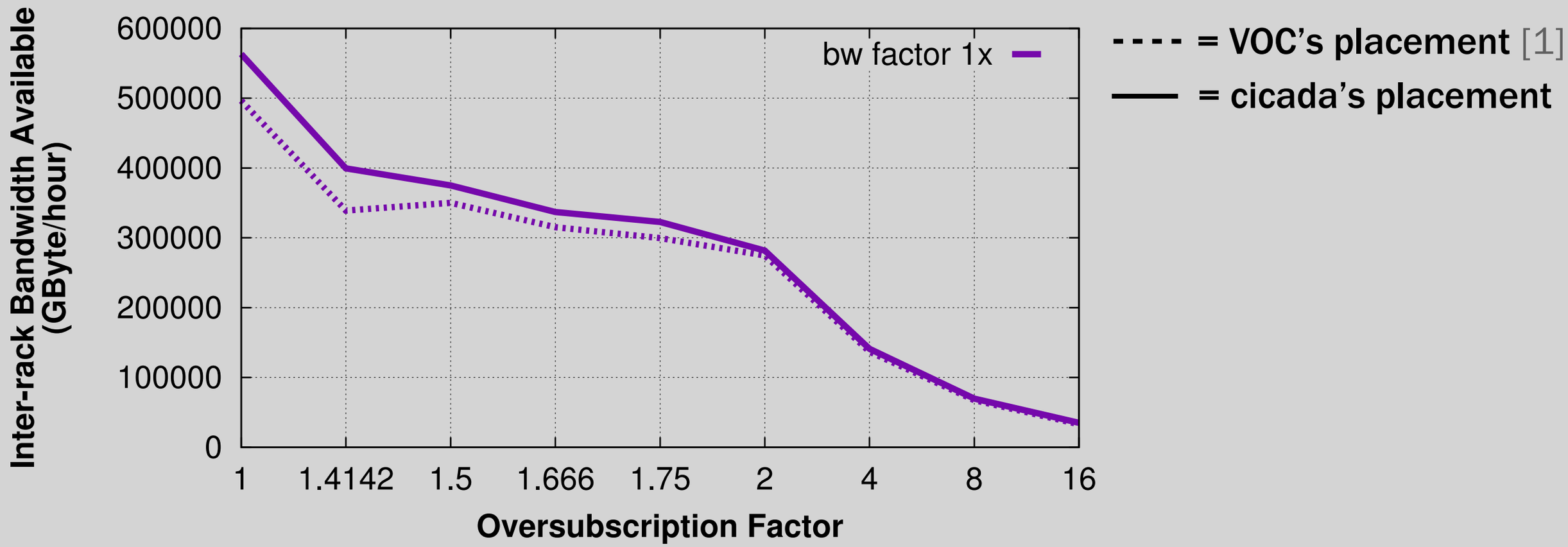


- - - - = VOC's placement [1]

[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

NETWORK UTILIZATION

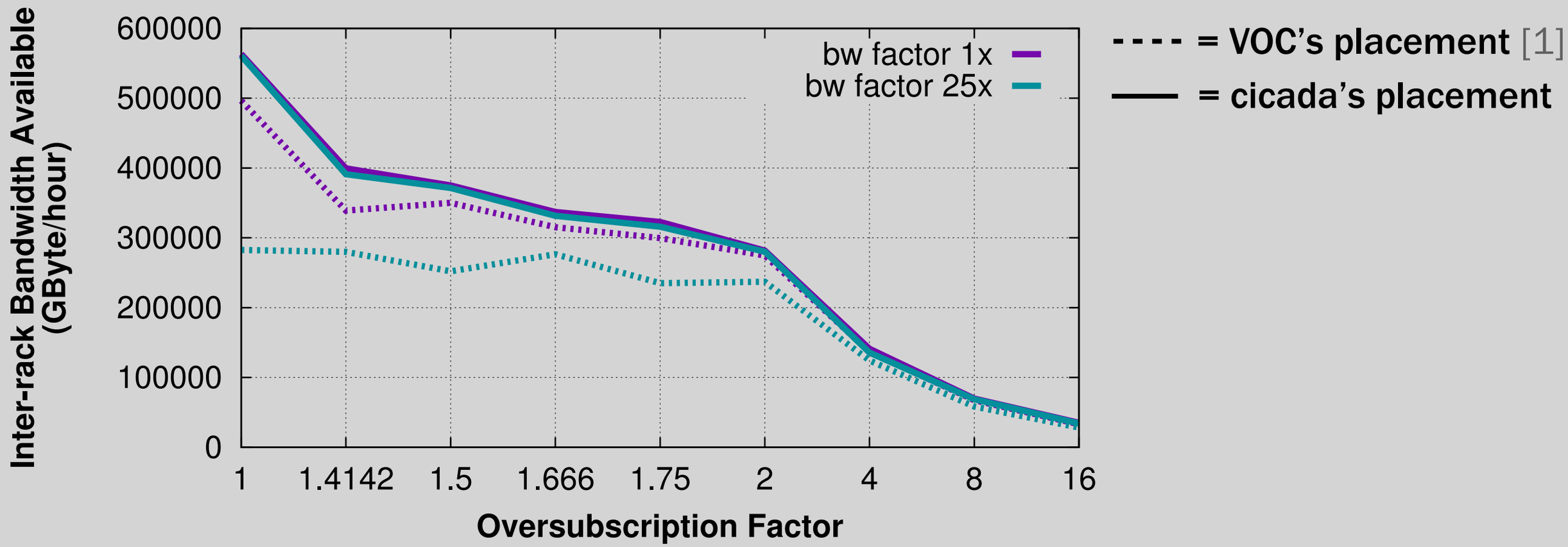
cicada's greedy placement heuristic: place the pairs of VMs that need the highest guarantees on the fastest paths (see paper for details)



[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

NETWORK UTILIZATION

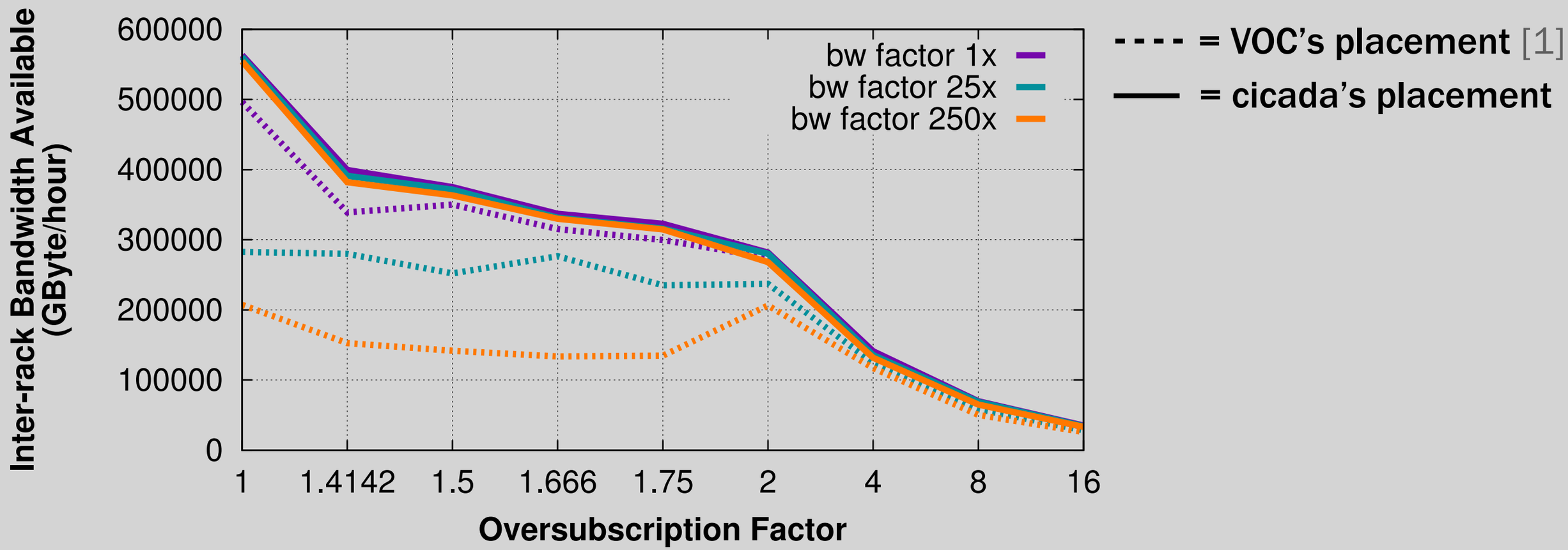
cicada's greedy placement heuristic: place the pairs of VMs that need the highest guarantees on the fastest paths (see paper for details)



[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

NETWORK UTILIZATION

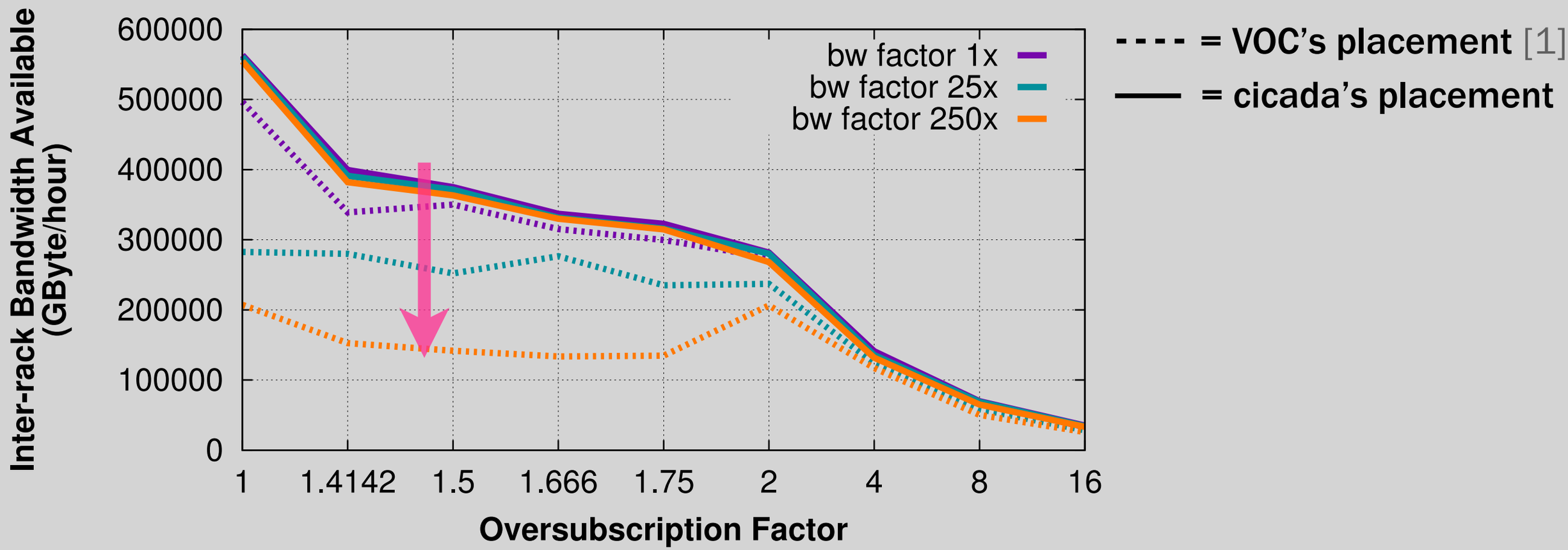
cicada's greedy placement heuristic: place the pairs of VMs that need the highest guarantees on the fastest paths (see paper for details)



[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

NETWORK UTILIZATION

cicada's greedy placement heuristic: place the pairs of VMs that need the highest guarantees on the fastest paths (see paper for details)

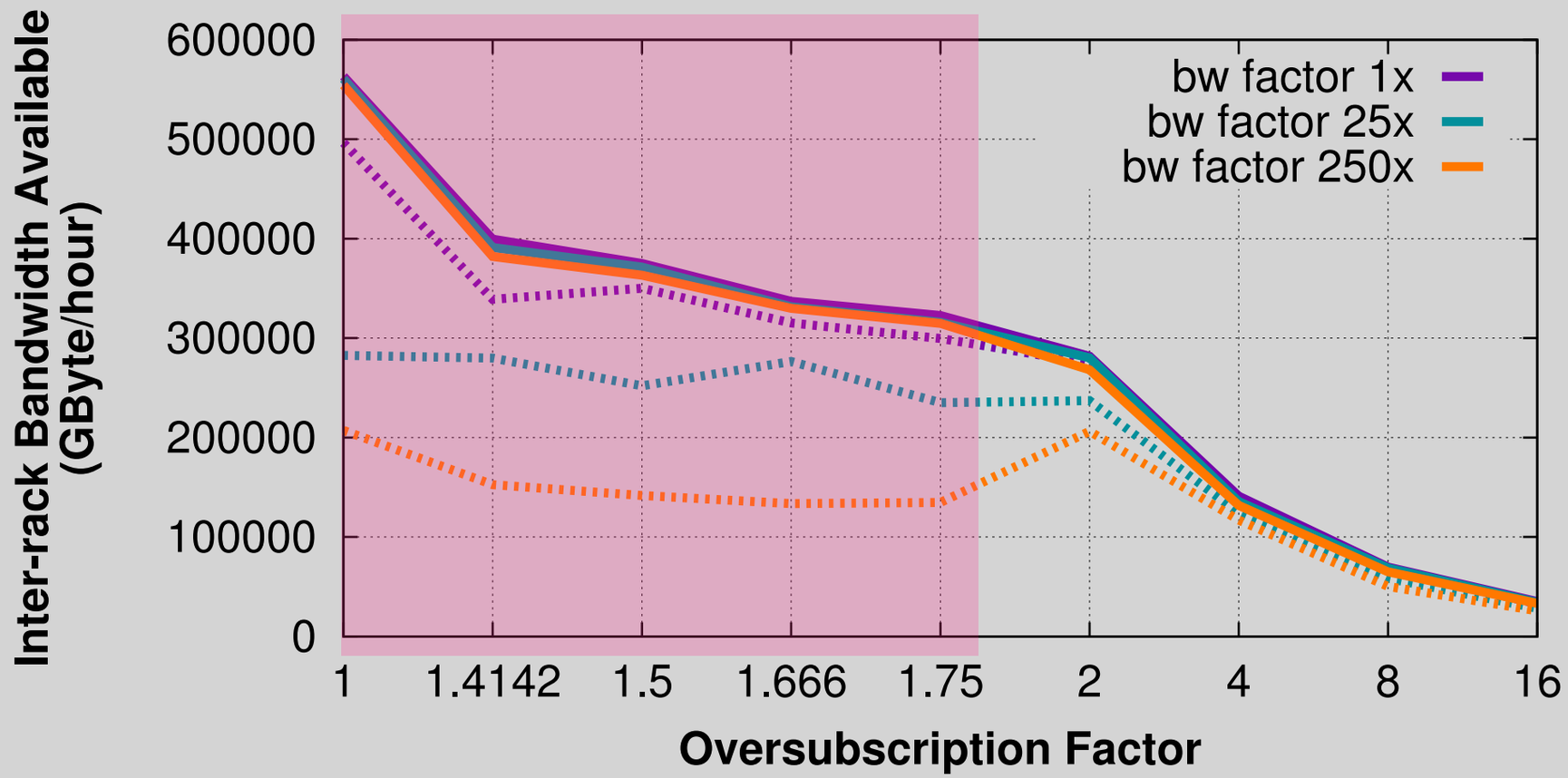


as applications use more bandwidth, VOC's placement wastes more inter-rack bandwidth

[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

NETWORK UTILIZATION

cicada's greedy placement heuristic: place the pairs of VMs that need the highest guarantees on the fastest paths (see paper for details)



- - - - = VOC's placement [1]

—— = cicada's placement

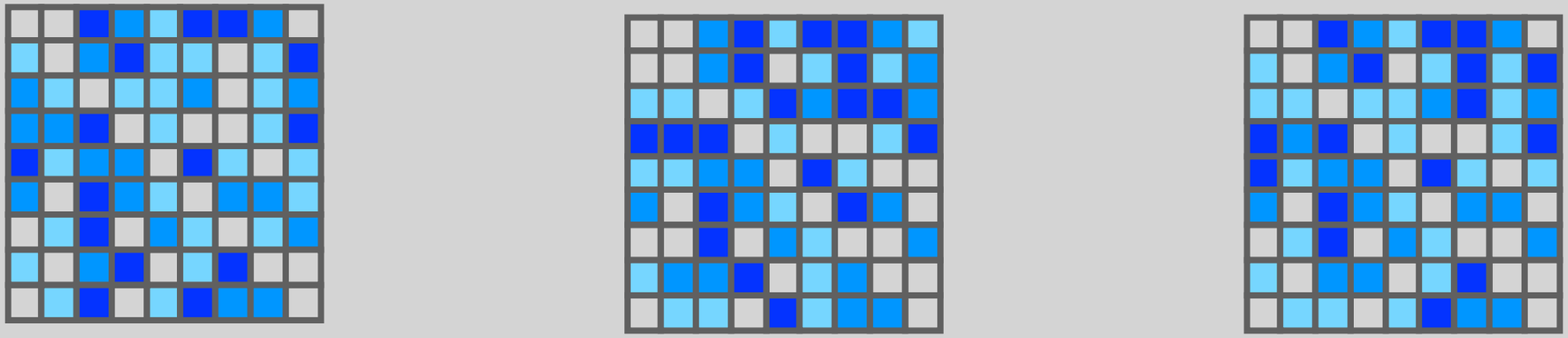
in some scenarios, cicada's placement more than doubles the amount of available inter-rack bandwidth

as applications use more bandwidth, VOC's placement wastes more inter-rack bandwidth

[1] Ballani, et al. "Towards Predictable Datacenter Networks". SIGCOMM 2011.

SUMMARY

cloud applications exhibit variability that existing models don't capture



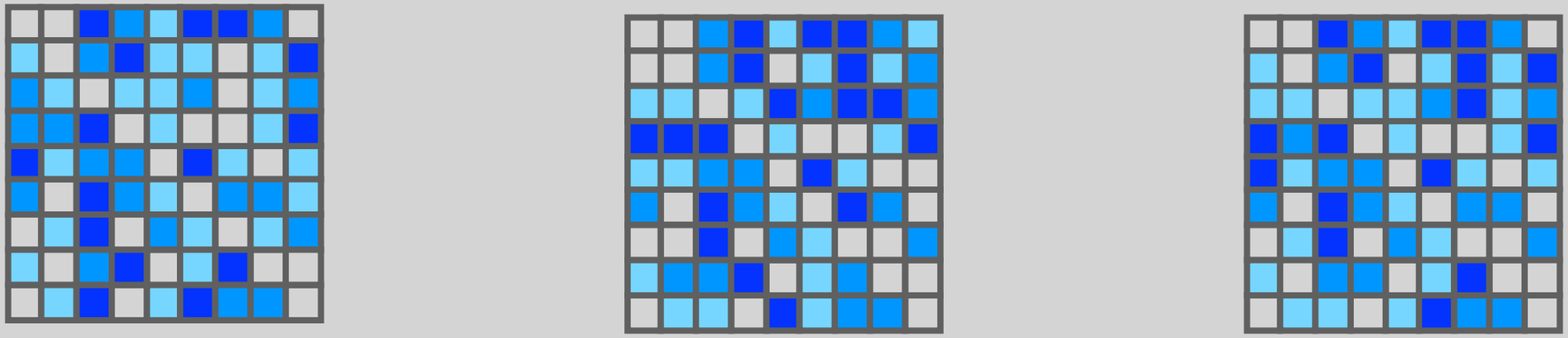
cicada captures this variability, and provides guarantees that are

- accurate
- calculated quickly*
- require little history
- increase network utilization

*see paper for this result

SUMMARY

cloud applications exhibit variability that existing models don't capture



cicada captures this variability, and provides guarantees that are

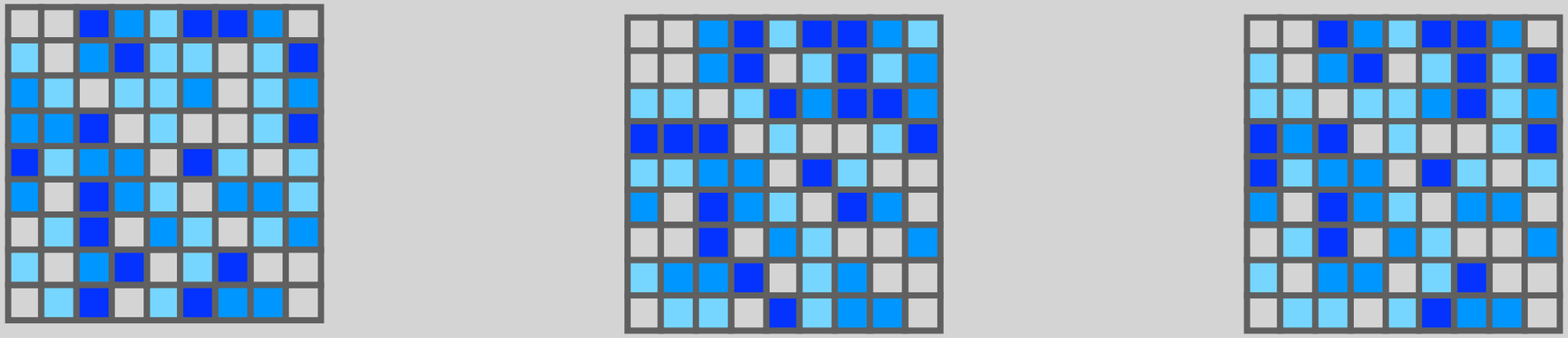
- accurate
- calculated quickly*
- require little history
- increase network utilization

improve relative error by 90%

*see paper for this result

SUMMARY

cloud applications exhibit variability that existing models don't capture



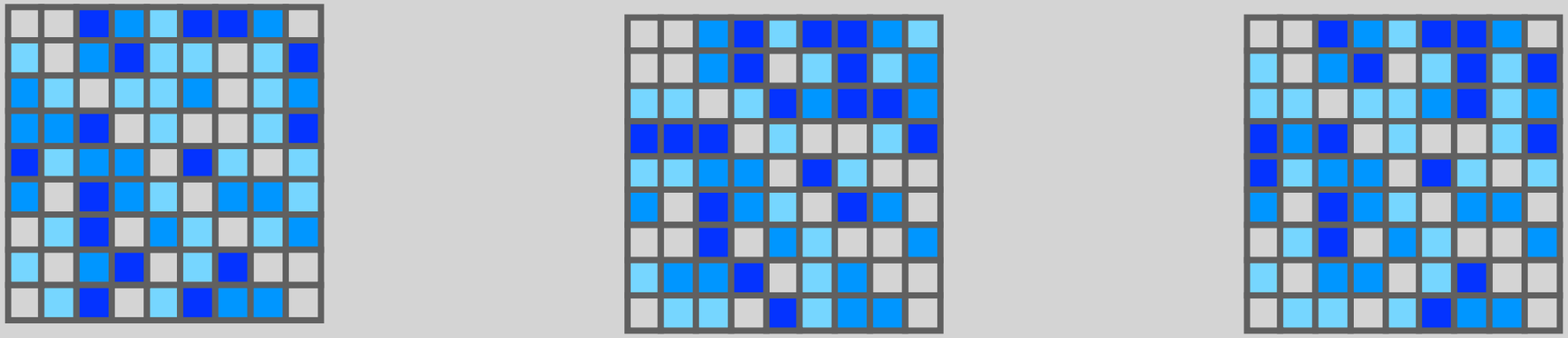
cicada captures this variability, and provides guarantees that are

accurate	improve relative error by 90%
calculated quickly*	< 10milliseconds per application
require little history	
increase network utilization	

*see paper for this result

SUMMARY

cloud applications exhibit variability that existing models don't capture



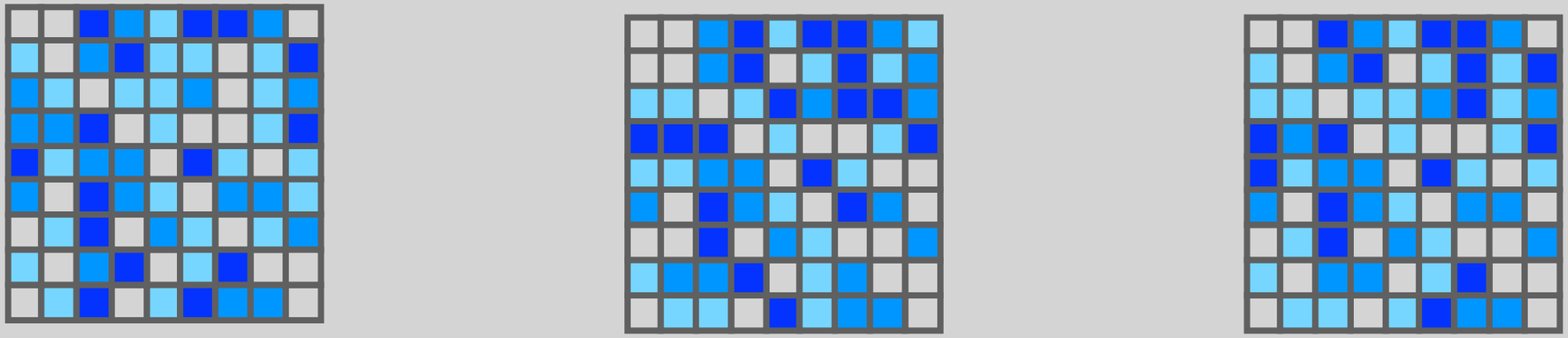
cicada captures this variability, and provides guarantees that are

accurate	improve relative error by 90%
calculated quickly*	< 10milliseconds per application
require little history	1–2 hours for most applications
increase network utilization	

*see paper for this result

SUMMARY

cloud applications exhibit variability that existing models don't capture



cicada captures this variability, and provides guarantees that are

accurate	improve relative error by 90%
calculated quickly*	< 10milliseconds per application
require little history	1–2 hours for most applications
increase network utilization	in some cases, inter-rack utilization is doubled

*see paper for this result