CICADA
PREDICTIVE GUARANTEES FOR CLOUD NETWORKS

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

10 machines
BANDWIDTH GUARANTEES

10 machines
BANDWIDTH GUARANTEES

10 machines
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?

WHY GUARANTEES?

aren’t cloud networks homogeneous and well-provisioned?

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

WHY GUARANTEES?

application traffic is affected by other users

aren’t cloud networks homogeneous and well-provisioned?

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

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?

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

POSSIBLE ARCHITECTURE
POSSIBLE ARCHITECTURE

customer requests machines and guarantee

10 machines, 10Gb/s between each pair
POSSIBLE ARCHITECTURE

customer requests machines and guarantee

10 machines, 10Gb/s between each pair

provider places customer’s VMs, enforces guarantee
POSSIBLE ARCHITECTURE

customer requests machines and guarantee

10 machines, 10Gb/s between each pair

provider places customer’s VMs, enforces guarantee

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

customer requests machines and guarantee

10 machines, 10Gb/s between each pair

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

10 machines, 10Gb/s between each pair

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

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

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

10 machines, 10Gb/s between each pair

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

10 machines, 10Gb/s between each pair

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

customer makes initial request for machines

provider places tenant VMs

hypervisors send measurements to cicada controller
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

hypervisors send measurements to cicada controller

cicada predicts bandwidth for each tenant
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

hypervisors send measurements to cicada controller

provider offers guarantee based on cicada’s prediction

cicada predicts bandwidth for each tenant
CICADA’S ARCHITECTURE

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

customer makes initial request for machines

cicada predicts bandwidth for each tenant

provider places tenant VMs

provider updates placement based on cicada’s prediction

hypervisors send measurements to cicada controller

provider offers guarantee based on cicada’s prediction
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

provider updates placement based on cicada’s prediction

hypervisors send measurements to cicada controller

provider offers guarantee based on cicada’s prediction

cicada predicts bandwidth for each tenant
CICADA’S ARCHITECTURE

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

- customer makes initial request for machines
- hypervisors send measurements to cicada controller
- provider places tenant VMs
- provider updates placement based on cicada’s prediction
- cicada predicts bandwidth for each tenant
- provider offers guarantee based on cicada’s prediction
- hypervisors send measurements to cicada controller
APPLICATION TRAFFIC

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

<table>
<thead>
<tr>
<th>VM1</th>
<th>VM2</th>
<th>VM3</th>
<th>VM4</th>
<th>VM5</th>
<th>VM6</th>
<th>VM7</th>
<th>VM8</th>
<th>VM9</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VM9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
to design cicada’s prediction algorithm, we need to understand how cloud applications send traffic

rigid application
(similar to VOC [1])

APPLICATION TRAFFIC

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

rigid application
(similar to VOC [1])

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

<table>
<thead>
<tr>
<th>vm1</th>
<th>vm2</th>
<th>vm3</th>
<th>vm4</th>
<th>vm5</th>
<th>vm6</th>
<th>vm7</th>
<th>vm8</th>
<th>vm9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

rigid application
(similar to VOC [1])

<table>
<thead>
<tr>
<th>vm1</th>
<th>vm2</th>
<th>vm3</th>
<th>vm4</th>
<th>vm5</th>
<th>vm6</th>
<th>vm7</th>
<th>vm8</th>
<th>vm9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vm9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

variable application

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

APPLICATION TRAFFIC

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

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

APPLICATION TRAFFIC

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

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

APPLICATION TRAFFIC

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

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

DATASET

---

HP Cloud Services
http://hpcloud.com
DATASET

HP Cloud Services
http://hpcloud.com

observed sflow data
(could also use tcpdump, etc.)
(in this talk)
collected one traffic matrix per hour, for each application
DATASET

HP Cloud Services
http://hpcloud.com

(observed sflow data
(could also use tcpdump, etc.)

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

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

$M_1$ $M_2$ ... $M_n$

observed traffic
**DATASET**

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

<table>
<thead>
<tr>
<th>$M_1$</th>
<th>$M_2$</th>
<th>...</th>
<th>$M_n$</th>
</tr>
</thead>
</table>

HP Cloud Services
http://hpcloud.com

observed sflow data (could also use tcpdump, etc.)

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?

- **no spatial variability**
- **some spatial variability**
- **high spatial variability**
SPATIAL VARIABILITY

how do we quantify 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?

1. let $F_{ij} = \text{fraction of tenant traffic sent from VM}_i \text{ to VM}_j$

no spatial variability

some spatial variability

high spatial variability

$F_{ij} = 1/20$
SPATIAL VARIABILITY

how do we quantify spatial variability?

1. let $F_{ij} = \text{fraction of tenant traffic sent from VM}_i \text{ to VM}_j$

no spatial variability

some spatial variability

high spatial variability

\[
\begin{align*}
F_{ij} &= \frac{1}{20} \\
\text{all } F_{ij} \text{ values are equal}
\end{align*}
\]
SPATIAL VARIABILITY

how do we quantify spatial variability?

1. let $F_{ij} = \text{fraction of tenant traffic sent from VM}_i \text{ to VM}_j$

**no spatial variability**

$$F_{ij} = \frac{1}{20}$$

all $F_{ij}$ values are equal

**some spatial variability**

$$F_{ij} = 0$$

**high spatial variability**

more data

less data

no data
SPATIAL VARIABILITY

how do we quantify spatial variability?

no spatial variability

some spatial variability

high spatial variability

1. let $F_{ij} = \text{fraction of tenant traffic sent from VM}_i \text{ to VM}_j$
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**
  - all $F_{ij}$ values are equal
  - $F_{ij} = 1/20$

- **some spatial variability**
  - two distinct $F_{ij}$ values
  - $F_{ij} = 0$, $F_{ij} = 1/12$

- **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

All $F_{ij}$ values are equal

$$F_{ij} = \frac{1}{20}$$

Some spatial variability

Two distinct $F_{ij}$ values

$$F_{ij} = 0 \quad F_{ij} = \frac{1}{12}$$

High spatial variability

More data

Less data

No data

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

1. let $F_{ij}$ = fraction of tenant traffic sent from VM$_i$ to VM$_j$

all $F_{ij}$ values are equal

some spatial variability

two distinct $F_{ij}$ values

high spatial variability

many distinct $F_{ij}$ values
**SPATIAL VARIABILITY**

how do we quantify spatial variability?

**no spatial variability**

- All $F_{ij}$ values are equal
- $F_{ij} = 1/20$

**some spatial variability**

- Two distinct $F_{ij}$ values
- $F_{ij} = 0$, $F_{ij} = 1/12$

**high spatial variability**

- Many distinct $F_{ij}$ values
- $F_{ij} < 1/20$, $F_{ij} > 1/20$

1. let $F_{ij} =$ fraction of tenant traffic sent from VM$_i$ to VM$_j$

2. calculate the coefficient of variation ($\sigma/\mu$) of the $F_{ij}$ values
SPATIAL VARIABILITY

how do we quantify spatial variability?

1. let $F_{ij} = \text{fraction of tenant traffic sent from VM}_i \text{ to VM}_j$

2. calculate the coefficient of variation ($\sigma/\mu$) of the $F_{ij}$ values

no spatial variability

- $F_{ij} = 1/20$
- all $F_{ij}$ values are equal
- $\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 0$

some spatial variability

- $F_{ij} = 0$
- two distinct $F_{ij}$ values

high spatial variability

- $F_{ij} < 1/20$
- $F_{ij} > 1/20$
- many distinct $F_{ij}$ values
**SPATIAL VARIABILITY**

how do we quantify spatial variability?

1. let $F_{ij} = \text{fraction of tenant traffic sent from VM}_i \text{ to VM}_j$
   
   **no spatial variability**
   
   \[
   F_{ij} = \frac{1}{20} \\
   \text{all } F_{ij} \text{ values are equal} \\
   \Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 0
   \]

2. calculate the coefficient of variation ($\sigma/\mu$) of the $F_{ij}$ values
   
   **some spatial variability**
   
   \[
   F_{ij} = 0 \\
   F_{ij} = \frac{1}{12} \\
   \text{two distinct } F_{ij} \text{ values} \\
   \Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 1^* \\
   \]

   **high spatial variability**
   
   \[
   F_{ij} < \frac{1}{20} \\
   F_{ij} > \frac{1}{20} \\
   \text{many distinct } F_{ij} \text{ values} \\
   \]

*see paper for this result
**SPATIAL VARIABILITY**

how do we quantify spatial variability?

1. let $F_{ij} =$ fraction of tenant traffic sent from $V M_i$ to $V M_j$

2. calculate the coefficient of variation ($\sigma/\mu$) of the $F_{ij}$ values

- **no spatial variability**
  - all $F_{ij}$ values are equal
  - $\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 0$

- **some spatial variability**
  - two distinct $F_{ij}$ values
  - $\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 1^*$

- **high spatial variability**
  - many distinct $F_{ij}$ values
  - $\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) > 1$

*see paper for this result*
SPATIAL VARIABILITY

how do we quantify spatial variability?

1. let $F_{ij}$ = fraction of tenant traffic sent from VM$_i$ to VM$_j$

- **no spatial variability**
  - all $F_{ij}$ values are equal
  - $\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 0$

- **some spatial variability**
  - two distinct $F_{ij}$ values
  - $\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) = 1^*$

- **high spatial variability**
  - many distinct $F_{ij}$ values
  - $\Rightarrow \sigma(F_{ij})/\mu(F_{ij}) > 1$

2. calculate the coefficient of variation ($\sigma/\mu$) of the $F_{ij}$ values

the higher the cov, the higher the spatial variation

$^*$see paper for this result
SPATIAL VARIABILITY

CDF

Spatial Variation

(coeff of $F_{ij}$ values)
applications exhibit **high** spatial variability
applications exhibit **high** spatial variability
applications exhibit high spatial variability
(the same result holds for temporal variability, see paper)
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 prediction algorithm draws inspiration from the following works:

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

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:

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:

Cicada's Algorithm

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:
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:
CICADA’S ALGORITHM

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

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

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

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.

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

\[
L(i,n) = \frac{\|\hat{M}_i - \hat{M}_n\|}{\|\hat{M}_n\|}
\]

Cicada’s prediction algorithm draws inspiration from the following works:
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 \textbf{online}; they are updated with every prediction based on errors made in previous predictions.

\[ w_i(n+1) = w_i(n) \cdot e^{-L(i,n)} \cdot \frac{1}{Z_{n+1}} \]

\[ L(i,n) = \frac{||\hat{M}_i - \bar{M}_n||}{||\bar{M}_n||} \]

Cicada’s prediction algorithm draws inspiration from the following works:
to evaluate cicada, we compared its predictions to ones made by a VOC-style system [1]

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)

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

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**

\[ E = \frac{||\hat{M} - M||}{||M||} \]

**relative error**

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

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

\[
L2\text{-norm error} \quad E = \frac{\|\hat{M} - M\|}{\|M\|}
\]

\[
\text{relative error} \quad E = \frac{\sum_{i=1}^{n^2} \hat{M}[i] - M[i]}{\sum_{i=1}^{n^2} M[i]}
\]

differentiates between over- and under-prediction

RESULTS  
PREDICTING AVERAGE DEMAND

does not differentiate between over- and under-prediction

differentiates between over- and under-prediction
RESULTS
PREDICTING AVERAGE DEMAND

Cicada’s predictions outperform VOC-style predictions (median error decreases by 90%) and require no customer input (71% for L2 error).

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

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

network resources must be available
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

accept cicada’s guarantee, but add 5% more bandwidth
cicada turns its predictions into pipe-model guarantees; a separate guarantee for each (source, destination) pair

network resources must be available

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

typically not confident for “small” applications

customers can add a buffer to the offered guarantees

accept cicada’s guarantee, but add 5% more bandwidth
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

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

typically not confident for “small” applications

observe packet drops
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
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?

intra-rack bandwidth is usually cheap (sometimes free)

inter-rack bandwidth is more expensive

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.

Intra-rack bandwidth is usually cheap (sometimes free).

Inter-rack bandwidth is more expensive.

Network diagram showing data flow and connections between racks.
cicada’s greedy placement heuristic: place the pairs of VMs that need the highest guarantees on the fastest paths (see paper for details)

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

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

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

---

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

---

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

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

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

[Grid image]

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
- Calculated quickly*
- Require little history
- Increase network utilization

*See paper for this result

Improves relative error by 90% in less than 10 milliseconds per application.
SUMMARY

cloud applications exhibit variability that existing models don’t capture

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

*see paper for this result

- improve relative error by 90%
- < 10 milliseconds per application
- 1–2 hours for most applications
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%< 10 milliseconds per application

1–2 hours for most applications

in some cases, inter-rack utilization is doubled

*see paper for this result