Virtual Network Embedding with Collocation
Benefits and Limitations of Pre-Clustering

Carlo Fürst\textsuperscript{1}, Stefan Schmid\textsuperscript{2}, Anja Feldmann\textsuperscript{1}

1: TU Berlin
2: TU Berlin & Telekom Innovation Laboratories

November 12, 2013
Today’s Datacenters...

- Multi-tenant virtualized
- Tenants typically pay for host resources
- Connectivity is guaranteed
Today’s Datacenters...

- Multi-tenant virtualized
- Tenants typically pay for host resources
- Connectivity is guaranteed

Problem [Ballani’11]:
Studies have shown that the intra-cloud bandwidth can vary by an order of magnitude.
⇒ Unpredictable application performance
Remove the uncertainty
Remove the uncertainty
Outline

- Explain model and problem
- Identify the impact of the collocation option on embedding algorithms
- Introduce *Pre-Clustering* - a technique to enable any existing algorithm to generate collocated embeddings
Virtual Network Embedding Problem

Physical Machine

Physical Link
Virtual Network Embedding Problem

- Physical Machine
  - Abstract aggregated
    "Compute Resource"

- Physical Link
  - Bandwidth
Virtual Network Embedding Problem
Virtual Network Embedding Problem

Virtual Node

Virtual Link

Virtual Node
Virtual Network Embedding Problem

Virtual Link
- Requested Compute Units

Virtual Node
- Requested Bandwidth
Virtual Network Embedding Problem
What is a ‘good’ mapping?
What is a ‘good’ mapping?
Existing Solutions

Many existing mapping algorithms

- ViNE [CHOWDHURY, Infocom 2009]
- SecondNet [GUO, Co-NEXT 2010]
- Oktopus [BALLANI, Sigcomm 2011]
- Isomorphism Detection [LISCHKA, Sigcomm 2009]
- Various Mixed-Integer-Programs
- ...
Existing Solutions

Diagram of network topology with nodes and connections.
Existing Solutions
Collocated Mappings

Physical Machine with capacity 2
Collocated Mappings
Benchmarking Algorithm: LoCo
Benchmarking Algorithm: LoCo
Benchmarking Algorithm: LoCo
Benchmarking Algorithm: LoCo
Benchmarking Algorithm: LoCo
Benchmarking Algorithm: LoCo

- Backtrack on failure
- Backtrack only over possible start nodes
- Graph exploration is directed by node / link resource requests
- Avoid Backtracking by forward checking
Evaluation Setup
Evaluation Setup

Add Requests

Until:
Sum of requested node resources = Sum of substrate node resources
## Evaluation Setup

<table>
<thead>
<tr>
<th>Request sequence</th>
<th>ADD REQ1</th>
<th>ADD REQ2</th>
<th>ADD REQ3</th>
<th>REM REQ1</th>
<th>ADD REQ4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Graph 1" /></td>
<td><img src="image2" alt="Graph 2" /></td>
<td><img src="image3" alt="Graph 3" /></td>
<td><img src="image4" alt="Graph 4" /></td>
<td><img src="image5" alt="Graph 5" /></td>
<td><img src="image6" alt="Graph 6" /></td>
</tr>
<tr>
<td>State</td>
<td><img src="image7" alt="State" /></td>
<td><img src="image8" alt="State" /></td>
<td><img src="image9" alt="State" /></td>
<td><img src="image10" alt="State" /></td>
<td><img src="image11" alt="State" /></td>
</tr>
</tbody>
</table>

Measure node utilization

Carlo Fürst (TU Berlin)  Virtual Network Embedding with Collocation  November 12, 2013
Evaluation Setup

<table>
<thead>
<tr>
<th>Request sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD REQ1</td>
</tr>
</tbody>
</table>

STATE

- Increase time until a Request expires
### Evaluation Setup

<table>
<thead>
<tr>
<th>Request sequence</th>
<th>ADD REQ1</th>
<th>ADD REQ2</th>
<th>ADD REQ3</th>
<th>REM REQ1</th>
<th>ADD REQ4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Graph1" /></td>
<td><img src="image2" alt="Graph2" /></td>
<td><img src="image3" alt="Graph3" /></td>
<td><img src="image4" alt="Graph4" /></td>
<td><img src="image5" alt="Graph5" /></td>
<td><img src="image6" alt="Graph6" /></td>
</tr>
</tbody>
</table>

Add Requests Until: ...

---

Carlo Fürst (TU Berlin) | Virtual Network Embedding with Collocation | November 12, 2013 | 10 / 21
Evaluation Setup

Request sequence

<table>
<thead>
<tr>
<th>ADD REQ1</th>
<th>ADD REQ2</th>
<th>ADD REQ3</th>
<th>REM REQ1</th>
<th>ADD REQ4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="FatTree" /></td>
<td><img src="image2" alt="FatTree" /></td>
<td><img src="image3" alt="FatTree" /></td>
<td><img src="image4" alt="FatTree" /></td>
<td><img src="image5" alt="FatTree" /></td>
</tr>
</tbody>
</table>

Substrate Topologies

- FatTree
Evaluation Setup

Request sequence

<table>
<thead>
<tr>
<th>ADD REQ1</th>
<th>ADD REQ2</th>
<th>ADD REQ3</th>
<th>REM REQ1</th>
<th>ADD REQ4</th>
</tr>
</thead>
</table>

Substrate Topologies

- FatTree

Embed. Algorithm

Unmodified Requests

- SNet
- LoCo
Impact of the collocation option

Slight Impact

Node Utilization

Loco  SecondNet
Impact of the collocation option

Slight Impact

Strong Impact

Node Utilization

Loco SecondNet

Loco SecondNet
Impact of the collocation option

Slight Impact

Strong Impact

Average Impact
Can we leverage the benefits of collocation with the existing algorithms?
Pre-Clustering
Pre-Clustering
Pre-Clustering
Pre-Clustering

We use:

- Farhat
- LoCo
- OptCut (runtime optimized MIP)
LoCo Preclustering
LoCo Preclustering
LoCo Preclustering
OptCut

- Generates an optimal (w.r.t. the amount of link resources between the merged nodes) Pre-Clustering
- Substrate is represented by two numbers:
  - $MAX_V$: The estimated host resources of a node
  - $MAX_E$: The estimated link resources attached to a node
- Run time independent of substrate size and topology
- Removes symmetry from the problem to speed up the solution process
Evaluation Parameters

Objective: Embed as many virtual resources as possible

**Substrate**
- DC topologies (default FatTree with 432 hosts)
- Each physical element has 4 resource units

**Requests**
- Randomized topologies (2-10 nodes, connection probability 0.15)
- Exponentially distributed duration with mean 10
- Resource sum of all requests \(\approx\) available substrate resources

All Per-Clustering approaches are combined with SecondNet
Experimental Pipeline

Request sequence

ADD REQ1  ADD REQ2  ADD REQ3  REM REQ1  ADD REQ4

Substrate Topologies

- FatTree
- BCube
- DCell

Pre-clustering

OptCut  Farhat  Loco

Embed. Algorithm

Modified Requests

SNet  SNet  SNet

Unmodified Requests

SNet  LoCo
Performance Analysis

All Pre-Clustering approaches improve the performance of SecondNet by factors > 1.5. But why is standalone LoCo in this scenario more performant?
Performance Analysis

All Pre-Clustering approaches improve the performance of Secondnet by factors $> 1.5$

But why is standalone LoCo in this scenario more preformant?
Reason I: Good Scenario for LoCo

Size 10

Node Utilization

LoCo  OptCut*  LoCo*  Farhat*  SNet

0.0  0.2  0.4  0.6  0.8  1.0
Reason I: Good Scenario for LoCo

Size 11

Node Utilization

LoCo  OptCut*  LoCo*  Farhat*  SNet

0.0  0.2  0.4  0.6  0.8  1.0

Carlo Fürst (TU Berlin)
Reason I: Good Scenario for LoCo

Size 12

Node Utilization

LoCo  OptCut*  LoCo*  Farhat*  SNet

0.0  0.2  0.4  0.6  0.8  1.0
Reason I: Good Scenario for LoCo

Size 13

Node Utilization

LoCo  OptCut*  LoCo*  Farhat*  SNet

Carlo Fürst (TU Berlin)  Virtual Network Embedding with Collocation  November 12, 2013  19 / 21
Reason I: Good Scenario for LoCo

Size 14

Node Utilization

LoCo  OptCut*  LoCo*  Farhat*  SNet

0.0  0.2  0.4  0.6  0.8  1.0
Reason 1: Good Scenario for LoCo

Size 15

Node Utilization

0.0 0.2 0.4 0.6 0.8 1.0

LoCo OptCut* LoCo* Farhat* SNet
Reason II: Fragmented Residual Resources

OptCut*

LoCo

Node Utilization

Load

Carlo Fürst (TU Berlin) Virtual Network Embedding with Collocation November 12, 2013 20 / 21
What else is in the paper?

- Description of the MetaTree Framework
What else is in the paper?

Contains
Represents
Meta Info
- Avail res.
- Interference
- Policy
- ...

Has

Description of the MetaTree Framework
Detailed description of LoCo
Concrete MIP formulations and evaluation
Runtime comparison
Impact of $\text{MAX}_E$ and $\text{MAX}_V$
What else is in the paper?

- Description of the MetaTree Framework
- Detailed description of LoCo
What else is in the paper?

Require: \( \text{VNet } G = (V, E), M = \{s\} \) for some \( s \in V(G), P = (\Gamma(s)) \)

\begin{algorithm}
\textbf{while} \( |P| > 0 \) \textbf{do}
  \begin{itemize}
  \item \textbf{sort} \( P \) (*) decreasing link capacities (*)
  \item \textbf{choose} \( u = P[0] \) (*) next node to map (*)
  \item \textbf{map} \( u \) (* forward checking *)
  \item \textbf{map} \( \{u, v\} \) \( \forall \ v \in M, \text{ where } \{u, v\} \in E(G) \)
  \item \( M = M \cup \{u\} \) \textbf{and} \( P = P \setminus \{u\} \)
  \end{itemize}
\textbf{end while}
\textbf{if} (embedding failed), \textbf{backtrack} on \( s \)
\end{algorithm}
What else is in the paper?

- Description of the MetaTree Framework
- Detailed description of LoCo
- Concrete MIP formulations and evaluation
  - Runtime comparison
  - Impact of $MAX_E$ and $MAX_V$
What else is in the paper?

### Constants:

- Set of nodes: \( V \)  
- Set of edges: \( E \subseteq V \times V \)  
- Weights: \( W : V \cup E \rightarrow \mathbb{R}^{\geq 0} \)  
- Maximal node resources: \( \text{MAX}_V \)  
- Maximal link resources: \( \text{MAX}_E \)  
- Larger nodes: \( \rho : V \rightarrow 2^V \)  

### Variables:

- Node mapping: \( \text{alloc}_V : V \times V \rightarrow \{0, 1\} \)  
- Auxiliary variable: \( x : E \times V \rightarrow \mathbb{R}^{\geq 0} \)
What else is in the paper?

- Description of the MetaTree Framework
- Detailed description of LoCo
- Concrete MIP formulations and evaluation
  - Runtime comparison
  - Impact of $MAX_E$ and $MAX_V$