SDN-based Virtual Machine Management for Cloud Data Centers

Richard Cziva  
David Stapleton  
Fung Po Tso  
Dimitrios P. Pezaros  

University of Glasgow  
University of Glasgow  
Liverpool John Moores University  
University of Glasgow
Agenda

• Motivation
• SDN suits for VM management
• A communication cost reduction scheme
• Design of our SDN-based VM management system
• Experimental results
• Conclusion
Motivation

In Cloud Data Centres, server and network resources have disjoint control mechanisms
Motivation

A **unified server-network control** mechanism is needed
Unified management of resources

Network

- Controller
  - network change notifications
  - migration orchestration

Servers

- Query flow statistics
- Get link weights for migration decision
In this paper...

• we propose a converged server-network control framework

• that exploits SDN to orchestrate live, network aware VM management

• to reduce the network-wide communication cost
S-CORE

• Scalable Communication Cost Reduction

Fung Po Tso, Konstantinos Oikonomou, Eleni Kavvadia, Dimitrios P. Pezaros
Scalable Traffic-Aware Virtual Machine Management for Cloud Data Centers
IEEE ICDCS 2014
communication cost for an allocations $A$

$$C(u, v) = \lambda(u, v) \sum_{i=1}^{\ell^A(u,v)} c_i.$$  

$\lambda(u,v)$ is the traffic load per time unit exchanged between VM $u$ and VM $v$  
link weight, $c$, can be set according to hierarchy, bandwidth, or policies but generally $c_1 < c_2 < c_3$  
$l(u,v)$ communication level between VM $u$ and VM $v$
Eventually, overall communication cost

\[ C^A = \sum_{u \in \mathcal{V}} \sum_{v \in \mathcal{V}} \lambda(u, v) \sum_{i=1}^{\ell^A(u,v)} c_i. \]

Thus, centralised optimal

\[ C^{opt} \leq C^A \]

\( \lambda(u,v) \) is the traffic load per time unit exchanged between VM \( u \) and VM \( v \)

link weight, \( c \), can be set according to hierarchy, bandwidth, or policies but generally \( c_1 < c_2 < c_3 \)

\( \ell(u,v) \) communication level between VM \( u \) and VM \( v \)
Limitations of S-CORE

- duplicates effort in measuring per-flow traffic load for each VM
- link costs are manually set
- network topology is manually set
- tokens for orchestration
SDN for VM management

The “Network” has all the information we need to calculate communication costs:

• link costs (levels)

• temporal usage

• topology

Let’s use SDN to get these information and orchestrate VM migration!
OpenFlow

Flow entry contains match rules, actions and stats
System design

- SDN controller (POX)
  - collecting flow statistics periodically (Statistics Request -> FlowStatsReceived)
  - managing topology, switches, hosts, link weights
  - orchestration of migration
- Hypervisors should support VM migration
Evaluation

• Mininet

• nping for traffic generation (static)
  • 50 byte TCP packets, 10 pps

• Two orchestration algorithms:
  • Round Robin
  • Load Aware
Evaluation

![Diagram of network levels and nodes]

**TABLE II.** INITIAL TRAFFIC GENERATION IN OUR TEST SETUP.

<table>
<thead>
<tr>
<th>Source VM</th>
<th>Source HV</th>
<th>Destination VM</th>
<th>Destination HV</th>
<th>Link cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>hv16</td>
<td>10.0.0.6</td>
<td>hv17</td>
<td>2</td>
</tr>
<tr>
<td>10.0.0.2</td>
<td>hv16</td>
<td>10.0.0.10</td>
<td>hv19</td>
<td>6</td>
</tr>
<tr>
<td>10.0.0.3</td>
<td>hv16</td>
<td>10.0.0.23</td>
<td>hv23</td>
<td>12</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>hv17</td>
<td>10.0.0.11</td>
<td>hv19</td>
<td>6</td>
</tr>
<tr>
<td>10.0.0.9</td>
<td>hv18</td>
<td>10.0.0.22</td>
<td>hv23</td>
<td>12</td>
</tr>
<tr>
<td>10.0.0.21</td>
<td>hv23</td>
<td>10.0.0.5</td>
<td>hv17</td>
<td>12</td>
</tr>
</tbody>
</table>
**Evaluation**

**Link cost: 12**

![Diagram](https://via.placeholder.com/150)

**TABLE II. INITIAL TRAFFIC GENERATION IN OUR TEST SETUP.**

<table>
<thead>
<tr>
<th>Source VM</th>
<th>Source HV</th>
<th>Destination VM</th>
<th>Destination HV</th>
<th>Link cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.1</td>
<td>hv16</td>
<td>10.0.0.6</td>
<td>hv17</td>
<td>2</td>
</tr>
<tr>
<td>10.0.0.2</td>
<td>hv16</td>
<td>10.0.0.10</td>
<td>hv19</td>
<td>6</td>
</tr>
<tr>
<td>10.0.0.3</td>
<td>hv16</td>
<td>10.0.0.23</td>
<td>hv23</td>
<td>12</td>
</tr>
<tr>
<td>10.0.0.6</td>
<td>hv17</td>
<td>10.0.0.11</td>
<td>hv19</td>
<td>6</td>
</tr>
<tr>
<td>10.0.0.9</td>
<td>hv18</td>
<td>10.0.0.22</td>
<td>hv23</td>
<td>12</td>
</tr>
<tr>
<td>10.0.0.21</td>
<td>hv23</td>
<td>10.0.0.5</td>
<td>hv17</td>
<td>12</td>
</tr>
</tbody>
</table>
Experimental Results

- Link utilisation

**Round Robin**

**Load Aware**

![Graphs showing link utilisation over the number of VM migrations for Round Robin and Load Aware strategies.](image)
Experimental Results

- **Link utilisation**

  VM1 migrated from hv16 -> hv17

  VM3 migrated from hv16 -> hv23
Experimental Results

- Link utilisation

![Link utilisation graphs]

- Round Robin
- Load Aware

still uses the core layer

end of core layer use
Experimental Results

- Overall communication cost
Future work

- Larger, more realistic experiments with OpenStack and OpenDaylight
- Dynamic traffic generation between VMs
- Stability improvements of the migration
Conclusion

- we presented a converged control plane that integrates server and network resource management
- SDN was used to calculate communication cost for each VM and we reallocate them to minimise the cost
Thank you for your attention

Richard Cziva - r.cziva.1@research.gla.ac.uk