2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing

# Inferring Network Topologies in Infrastructure as a Service Cloud

Dominic Battré*, Natalia Frejnik*, Siddhant Goel†, Odej Kao* and Daniel Warneke*

*Technische Universität Berlin, Berlin, Germany
Email: {firstname}.{lastname}@tu-berlin.de

†Technische Universität München, Munich, Germany
Email: goel@in.tum.de

*Abstract*—**Infrastructure as a Service (IaaS) clouds are gaining increasing popularity as a platform for distributed computations. The virtualization layers of those clouds offer new possibilities for rapid resource provisioning, but also hide aspects of the underlying IT infrastructure which have often been exploited in classic cluster environments. One of those hidden aspects is the network topology, i.e. the way the rented virtual machines are physically interconnected inside the cloud.**

**We propose an approach to infer the network topology connecting a set of virtual machines in IaaS clouds and exploit it for data-intensive distributed applications. Our inference approach relies on delay-based end-to-end measurements and can be combined with traditional IP-level topology information, if available. We evaluate the inference accuracy using the popular hypervisors KVM as well as XEN and highlight possible performance gains for distributed applications.**

*Keywords*-**Cloud Computing, Topology Inference**

## I. Introduction

Recently, cloud computing has experienced a growing interest as a platform for flexible, large-scale applications. Operators of so-called Infrastructure as a Service (IaaS) clouds, e.g. Amazon EC2 [1], let their customers allocate and control sets of virtual machines (VMs) and charge them on a pay-as-you-go basis. The VMs themselves are hosted inside the operator's data center without exposing any details about the underlying IT infrastructure to the customer.

While this level of abstraction simplifies the deployment of VMs significantly, it comes at the expense of losing topology information, i.e. information on how the rented VMs are physically interconnected. In particular data-intensive distributed applications, e.g. Apache Hadoop [2], offer to take the network topology into account in order to exploit data locality and reduce the risk of network bottlenecks [3].

This research presents an approach to reconstruct likely network topologies connecting a set of VMs in those so far opaque IaaS clouds. Following the idea of network tomography [4], our approach relies on end-to-end measurements. Thereby the VMs exchange a series of probe packets to determine the characteristics of the network links which connect them. By correlating these link characteristics, it becomes possible to infer (parts of) the underlying network topology and exploit this knowledge at the application level.

The remainder of this paper briefly sketches the design and implementation of our topology inference approach and provides an overview of our experimental results using the popular open source hypervisors KVM [5] and XEN [6].

## II. Design and Implementation

Our topology inference approach is designed to work on a set of VMs which are connected through a tree-like, but initially unknown network structure.

Initially, one VM from this set is elected as a master node and starts sending out ICMP echo requests to all other VMs. The goal of this operation is to obtain a first coarse-grained IP-level network topology. Based on this IP-level network topology, we subdivide the overall set of VMs into subsets of VMs which are in the same IP subnet (Figure 1). If (parts of) the IP-level network topology could not be obtained, e.g. because ICMP echoes are disabled by the cloud operator, a subset may also include VMs from different IP subnets.

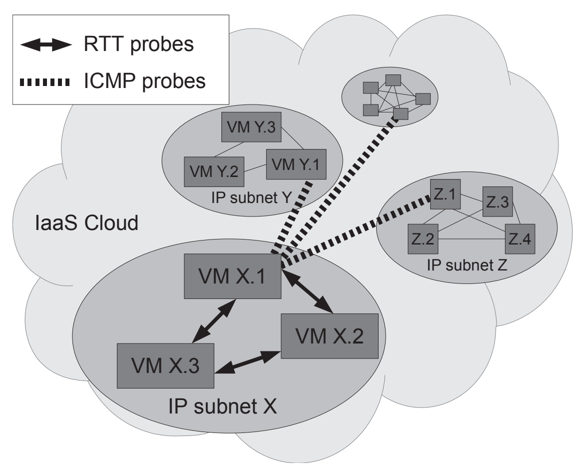


Figure 1. Schematic overview of the topology inference approach

In each subset we then start the end-to-end measurements. The goal of the end-to-end measurements is to detect internal network components which operate underneath the IP layer, like e.g. link layer switches or bridges which connect the individual VMs to the physical network.

978-0-7695-4395-6/11 $26.00 © 2011 IEEE
DOI 10.1109/CCGrid.2011.79

Through extensive experimental evaluations with KVM as well as XEN we found round-trip time (RTT) probes which are exchanged between each pair of VMs within a subset to provide the most robust proximity metric for the inference process in presence of hardware virtualization. In particular, we observed a distinct gap in RTTs between co-located VMs, i.e. VMs being hosted on the same physical server, and those running on different servers. The RTT gap grew to several milliseconds with increasing background traffic using both paravirtualized KVM and XEN VMs.

In general, we found the virtualization layer to have a significant impact on the observable link characteristics, E.g., when we used full virtualization instead of paravirtualization, the measured delay between the individual VMs showed large variations which can decrease the accuracy of the inferred topologies tremendously.

Based on RTTs as a proximity metric between each pair of VMs, our approach uses a bottom-up clustering algorithm for the actual topology inference within a subset [7]. Starting with each VM as an individual cluster, the algorithm progressively merges clusters with the closest proximity until only one cluster is left. The algorithm is executed by one VM in each subset. Eventually, the inferred network topology of each subset is reported back to the master node which uses the information to refine the global IP-level topology.

## III. Experimental Evaluation

We evaluated the accuracy of the inferred topologies on our local cloud testbed. The testbed consisted of 64 VMs (KVM, paravirtualization) running on eight physical servers, which were all connected to a central Ethernet switch. We express the accuracy of the inferred topology as the mean Jaccard similarity between the sets of inferred co-located VMs and the actually co-located ones. During the experiments we generated different levels of background traffic. The results are shown in Table I.

| Set Sim. | Background Traffic among VMs in MBit/s | | | | | | |
|---|---|---|---|---|---|---|---|
| | 0 | 100 | 200 | 300 | 400 | 500 | 600 |
| XEN | 0.95 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| KVM | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Table I
MEAN SIMILARITY OF INFERRED SETS OF CO-LOCATED VMS AND ACTUAL ONES

To highlight the potential benefit of the inferred network topologies for cloud applications, we integrated our topology inference approach into our parallel data processing framework Nephele [8]. The sample job we devised consisted of eight data-parallel producer tasks sending a total of 200 GB of data to eight consumer tasks. Each task was scheduled to run on a separate VM. The results of the experiment indicate that topology-aware scheduling based on the inferred network topology led to a speedup by approx. factor 2 compared to random task placement and by approx. factor 4 compared to worst-case task placement.

## IV. Conclusion and Future Work

As a result of our research we can conclude that network topology inference in IaaS clouds is a challenging, but with respect to data-intensive distributed applications also rewarding subject.

On the one hand end-to-end measurements allow for a reliable detection of co-located VMs in paravirtualized environments due to the significant differences in the RTTs. On the other hand the variations introduced by the virtualization layer in terms of packet delay are currently so high that there seems to be little potential to reliably infer passive network components like link layer switches.

In general, we think our work represents a valuable contribution to the current efforts of porting data-intensive distributed applications to the cloud. For future work we are curious to follow new developments in the field of virtualization, especially with respect to hardware-assisted I/O virtualization.

## Acknowledgment

We thank HP for supporting this work through an Open Collaboration Grant.

## References

[1] Amazon Web Services LLC, "Amazon Elastic Compute Cloud (Amazon EC2)," http://aws.amazon.com/ec2/, 2010.

[2] The Apache Software Foundation, "Welcome to Hadoop!" http://hadoop.apache.org/, 2010.

[3] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[4] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: recent developments," *Statistical Science*, vol. 19, pp. 499–517, 2004.

[5] A. Kivity, "kvm: the Linux virtual machine monitor," in *OLS '07: The 2007 Ottawa Linux Symposium*, July 2007, pp. 225–230.

[6] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2003, pp. 164–177.

[7] J. A. Hartigan, *Clustering Algorithms*. John Wiley & Sons, 1975.

[8] D. Warneke and O. Kao, "Nephele: Efficient parallel data processing in the cloud," in *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers*, ser. MTAGS '09. New York, NY, USA: ACM, 2009, pp. 8:1–8:10.