Optimizing Long-Lived CloudNets with Migrations

Gregor Schaffrath, Stefan Schmid, Anja Feldmann
November, 2012
Optimizing Long-Lived CloudNets with Migrations

Gregor Schaffrath, Stefan Schmid, Anja Feldmann

November, 2012
Optimizing Long-Lived CloudNets with Migrations

Gregor Schaffrath, Stefan Schmid, Anja Feldmann
November, 2012
Cloud computing is a big success!
But what is the point of clouds if they cannot be accessed?

Network matters!
Next Natural Step for Virtualization!

**Success of Node Virtualization**
revamped server business

**Trend of Link Virtualization**
- Predictable execution times (e.g., cheaper executions)
- Isolation and QoS
- E.g., VLANs, Software Defined Networks (SDN) / OpenFlow, ...

Unified, fully virtualized networks: **CloudNets**

„Combine networking with heterogeneouse cloud resources (e.g., storage, CPU, ...)!“
Next Natural Step for Virtualization!

Trend of Link Virtualization

In this talk:

CloudNets not only for data centers but also for widearea networks

...
A Use Case: Specify network, not only VMs!

Connecting Providers (Geographic Footprint).

**CloudNet 1: Computation**
Specification:
1. > 1 GFLOPS per node
2. Monday 3pm-5pm
3. multi provider ok

**CloudNet 2: Mobile service w/ QoS**
Specification:
1. close to mobile clients
2. >100 kbit/s bandwidth for synchronization

**CloudNet requests**

**Provider 1**

**Provider 2**

**Physical infrastructure (e.g., accessed by mobile clients)**
A Use Case: Specify network, not only VMs!

Connecting Providers (Geographic Footprint).

**CloudNet 1: Computation**

Specification:
1. > 1 GFLOPS per node
2. Monday 3pm - 5pm
3. multi provider ok

**CloudNet 2: Mobile service w/ QoS**

Specification:
1. close to mobile clients

**Benefits**
- ISP: new services, QoS VPN, ...
- Startups: no own infrastructure needed
- Datacenters: meet deadlines (D3/Octopus, SecondNet, ...)
- «Virtual» = can migrate (links and nodes!)

Physical infrastructure (e.g., accessed by mobile clients)
The Prototype: Embedding and Seamless Migration.

- Open vSwitch supports VLAN bridging
- To VM looks like Ethernet (no VLAN)
- Wide-area: open VPN tunnel

- Each virtual link is a VLAN (broadcast domain)
- Migration: reconfigure VLANs, not addresses of virtual nodes!
- Transparent for users...

Migration
Happens at various stages!

Business roles: can map CloudNet on CloudNet (not only bare metal)
How to Embed CloudNets Efficiently?

Computationally hard...
Our 2-stage approach:

Stage 1: Map quickly and heuristically (dedicated resources)

Stage 2: Migrate long-lived CloudNets to «better» locations (min max load, max free resources, ...)
Typically: heavy-tailed durations, so old CloudNets will stay longer!
Communicate CloudNets, substrate resources and embeddings to business partners or customers:

Resource description language

Input for Embedding Algo: CloudNet Spec.

ICCCN 2012
Exploiting Flexibilities: Resource Description Language.

For example: Web service with two virtual nodes (connected for synchronization)

Given a CloudNet specification, how to realize/embed the network?

Goal: Respect specifications, but do not impose any additional constraints! (Maintain specification / virtualization flexibility.)
General Mathematical Program (MIP)

Nodes:
- map_node: \[ \sum_{v \in N_{E}} \text{new}(u, v) = 1 \quad u \in N_{E_{V}} \]
- set_new: \[ \text{alloc}_{N_{E}}(u, v, v) \leq \text{cap}_{N_{E}}(u) \cdot \text{new}(u, v) \quad v \in N_{E_{V}} \quad u \in N_{E_{D}} \]
- reg_min: \[ \text{alloc}_{N_{E}}(u, v, v) \geq \text{new}(u, v) \cdot \text{req}(u, v, s) \quad u \in N_{E_{V}} \quad v \in N_{E_{D}} \]
- reg_max: \[ \text{alloc}_{N_{E}}(u, v, v) \leq \text{new}(u, v) \cdot \text{req}(u, v, s) \quad u \in N_{E_{V}} \quad v \in N_{E_{D}} \]
- reg_com: \[ \text{alloc}_{N_{E}}(u, v, v) = \text{new}(u, v) \cdot \text{req}(u, v, s) \quad u \in N_{E_{V}} \quad v \in N_{E_{D}} \]

Mapping:
- relate_u: \[ \text{alloc}_{N_{E}}(u, v, v) \geq \min \text{alloc}_{N_{E}}(u, v, v) \quad u \in N_{E_{V}} \quad v \in N_{E_{D}} \]
- max_load: \[ \text{load}(u, v, v) \leq \text{max_load} \quad u \in N_{E_{V}} \quad v \in N_{E_{D}} \]

Resource-Variable Relation:
- resource: \[ \sum_{v \in N_{E}} \text{prop}(v, v) \cdot \text{alloc}_{N_{E}}(u, v, v) = \text{alloc}_{N_{E}}(u, v, v) \quad v \in N_{E_{V}} \quad u \in N_{E_{D}} \]

Links:
- map_link: \[ \sum_{v \in N_{E}} \text{new}(u, v) \geq 1 \quad u \in N_{E_{V}} \]
- map_flow: \[ \text{new}(f, u) \geq \text{new}(u, v) \quad \forall f \in F(u), \forall v \in N_{E_{D}}, \forall u \in N_{E_{V}} \]
- map_src: \[ \text{new}(f, u) \geq \text{new}(u, v) \quad \forall f \in F(u), \forall v \in N_{E_{D}}, \forall u \in N_{E_{V}} \]
- map_sink: \[ \text{new}(f, u) \geq \text{new}(u, v) \quad \forall f \in F(u), \forall v \in N_{E_{D}}, \forall u \in N_{E_{V}} \]
- reg_link: \[ \sum_{v \in N_{E}} \text{flow}_{L}(f, u, v) \cdot \text{flow}_{N}(f, u, v) = \text{flow}_{E}(f, u, v) \quad v \in N_{E_{V}} \quad u \in N_{E_{D}} \]
- reg_max: \[ \sum_{v \in N_{E}} \text{flow}_{L}(f, u, v) \cdot \text{flow}_{N}(f, u, v) \leq \text{new}(f, v) \cdot \text{req}(u, v, s) + \text{new}(f, v) \cdot \text{req}(u, v, s) \quad v \in N_{E_{V}} \quad u \in N_{E_{D}} \]
- reg_const: \[ \sum_{v \in N_{E}} \text{flow}_{L}(f, u, v) \cdot \text{flow}_{N}(f, u, v) = \text{new}(f, v) \cdot \text{req}(u, v, s) + \text{new}(f, v) \cdot \text{req}(u, v, s) \quad v \in N_{E_{V}} \quad u \in N_{E_{D}} \]

Link Allocation:
- exp_out: \[ \sum_{v \in N_{E}} \text{flow}_{L}(f, u, v) \leq \text{alloc}_{N_{E}}(u, v, v) \quad \forall f \in F(u), \forall v \in N_{E_{D}}, \forall u \in N_{E_{V}} \]
- exp_in: \[ \sum_{v \in N_{E}} \text{flow}_{L}(f, u, v) \leq \text{alloc}_{N_{E}}(u, v, v) \quad \forall f \in F(u), \forall v \in N_{E_{D}}, \forall u \in N_{E_{V}} \]
- direction: \[ \text{flow}_{L}(f, u, v) \leq \text{new}(u, v) \cdot \text{cap}_{N_{E}}(u, v) \quad \forall f \in F(u), \forall v \in N_{E_{D}}, \forall u \in N_{E_{V}} \]
- relate_f: \[ \sum_{v \in N_{E}} \text{flow}_{L}(f, u, v) \leq \text{new}(u, v) \cdot \text{cap}_{N_{E}}(u, v) \quad \forall f \in F(u), \forall v \in N_{E_{D}}, \forall u \in N_{E_{V}} \]

Advantages:
1. Generic (backbone vs datacenter) and allows for migration
2. Allows for different objective functions
3. Optimal embedding: for background optimization of heavy-tailed (i.e., long-lived) CloudNets, quick placement e.g., by clustering

But: slow...
The Solution.

General Mathematical Program (MIP)

Advantages of MIP:

- Very general
- Supports easy replacement of objective functions (load balancing vs load concentration)
- Can use standard, optimized software tools such as CPLEX, Gorubi, etc.

Advantages:

1. Generic (backbone vs datacenter) and allows for migration
2. Allows for different objective functions
3. Optimal embedding: for background optimization of heavy-tailed (i.e., long-lived) CloudNets, quick placement e.g., by clustering

Schaffrath et al.: UCC 2012
Generality of the MIP.

Objective functions:
- minimize maximum load (= load balance)
- maximize free resources (= compress as much as possible), ...
- answer questions: «is it worthwhile to unbalance now to save energy?»

Migration support:
- costs for migration: per element, may depend on destination, etc.
- answer questions such as «what is cost/benefit if I migrate now?»

Embedding:
- embedding full-duplex on full-duplex links
- full-duplex on half-duplex links
- or even multiple endpoint links (e.g., wireless) supported!
What is the Use of Flexibility?

PoS

How much link resources are needed to embed a CloudNet with specificity s%?

Up to 60%, even a little bit more if no migrations are possible!

Skewed (Zipf) distributions worst when not matching.
What is the Use of Flexibility?

PoS

How much link resources needed to embed a CloudNet with specificity s%?

Ludwig et al.: UCC 2012
On the Use of Migration.

Migration: Useful to increase the number of embeddable CloudNets, especially in under-provisioned scenarios


**Performance of the MIP: Setup.**

**Substrate:** Rocketfuel ISP topologies (with 25 nodes)

**CloudNets:** Out-sourcing scenario, CloudNets with up to ten nodes, subset of nodes fixed (access points) and subset flexible (cloud resources)

**Solver:** CPLEX on 8-core Xeon (2.5GHz)
Performance of the MIP.

- Runtime **below 1 minute** per CloudNet, slightly increasing under load
- Impact of **CloudNet size** relatively small
Performance of the MIP.

- Enabling option to migrate can increase execution time significantly (log scale!)
- Also number of flexible CloudNet components is important
What about time?
Basis for online embeddings!

Even et al.:
ICDCN 2012 (best paper)
Supported Traffic Models.

**Customer Pipe**
Every pair \((u,v)\) of nodes requires a certain bandwidth.

Detailed constraints, only this traffic matrix needs to be fulfilled!

**Hose Model**
Each node \(v\) has max ingress and max egress bandwidth: each traffic matrix fulfilling them must be served.

More flexible, must support many traffic matrices!

**Aggregate Ingress Model**
Sum of ingress bandwidths must be at most a parameter \(I\).

Simple and flexible! Good for multicasts etc.: no overhead, duplicate packets for output links, not input links already!
Supported Routing Models.

**Tree**
VNet is embedded as Steiner tree:

**Single Path**
Each pair of nodes communicates along a single path.

**Multi Path**
A linear combination specifies split of traffic between two nodes.
Conclusion.

- **Trend towards virtualization & elastic networking**
  - cloud extends to network
  - **CloudNets**: connecting cloud resources with virtual networking

- **Embedding CloudNets a major challenge**
  - Heterogenous environment: datacenter vs access network vs backbone, VLANs vs OpenFlow vs MPLS, placement policies, ...

- **Our algorithm is very generic...**

- ... **but embeddings still relatively fast**
  - compare, e.g., to time to request an MPLS topology today?
  - Or to time of large Map Reduce jobs?

*Future work: tempo, tempo, tempo 😊*
Conclusion.

- **Trend towards virtualization & elastic networking**
  - cloud extends to network
  - **CloudNets**: connecting cloud resources with virtual networking

- **Our algorithm is very generic...**

- **... but embeddings still relatively fast**
  - compare, e.g., to time to request an MPLS topology today?
  - Or to time of large Map Reduce jobs?

**Future work:** **tempo, tempo, tempo 😊**
Combining Clouds with Virtual Networking

The CloudNet Project

Internet Network Architectures (INET)
TU Berlin / Telekom Innovation Labs (T-Labs)
Contact: Stefan Schmid

News
Overview
People
Publications
Demo
Talks/Posters

News

- Watch on YouTube: migration demonstrator video!
- We are looking for students and interns with good algorithmic background to contribute to Virtul. Contact us for more details or have a look at some open topics.

Project Overview

CloudNets are virtual networks (VNets) connecting cloud resources. The network virtualization paradigm allows to run multiple CloudNets on top of a shared physical infrastructure. These CloudNets can have different properties (provide different security or QoS guarantees, run different protocols, etc.) and can be managed independently of each other. Moreover, parts of a CloudNet can be migrated dynamically to locations where the service is most useful or most cost efficient (e.g., in terms of energy conservation). Depending on the circumstances and the technology, these migrations can be done live and without interrupting ongoing sessions. The flexibility of the paradigm and the decoupling of the services from the underlying resource networks has many advantages; for example, it facilitates a more efficient use of the given resources, it promises faster innovations by overcoming the ossification of today's Internet architecture, it simplifies the network management, and it can improve service performance.

We are currently developing a prototype system for this paradigm (currently based on VLANs), which raises many scientific challenges. For example, we address the problem of how to embed CloudNet requests (e.g., see [1] for online CloudNet embeddings and [2] for a general mathematical embedding program), or devise algorithms to migrate CloudNets to new locations (e.g., due to user mobility) taking into account the...
Collaborators and Publications.

• People
  • **T-Labs / TU Berlin:** Anja Feldmann, Carlo Fürst, Johannes Grassler, Arne Ludwig, Matthias Rost, Gregor Schaffrath, Stefan Schmid
  • **Uni Wroclaw:** Marcin Bienkowski
  • **Uni Tel Aviv:** Guy Even, Moti Medina
  • **NTT DoCoMo Eurolabs:** Group around Wolfgang Kellerer

• Publications
  • **Prototype:** VISA 2009, ERCIM News 2012, ICCCN 2012
  • **Migration:** VISA 2010, IPTComm 2011, HotICE 2011
  • **Embedding:** 2 x UCC 2012, DISC 2012, ICDCN 2012 (*Best Paper Distributed Computing Track*)
Contact.

Dr. Stefan Schmid

Telekom Innovation Laboratories
Ernst-Reuter-Platz 7, D-10587 Berlin
E-mail: stefan@net.t-labs.tu-berlin.de
Project website:
http://www.net.t-labs.tu-berlin.de/~stefan/virtu.shtml