A network transparent solution for flexibly provisioning connected virtual infrastructure across multiple data centers

Huan Zhou, Paul Martin, Cees de Laat, Zhiming Zhao
University of Amsterdam, Amsterdam, the Netherlands

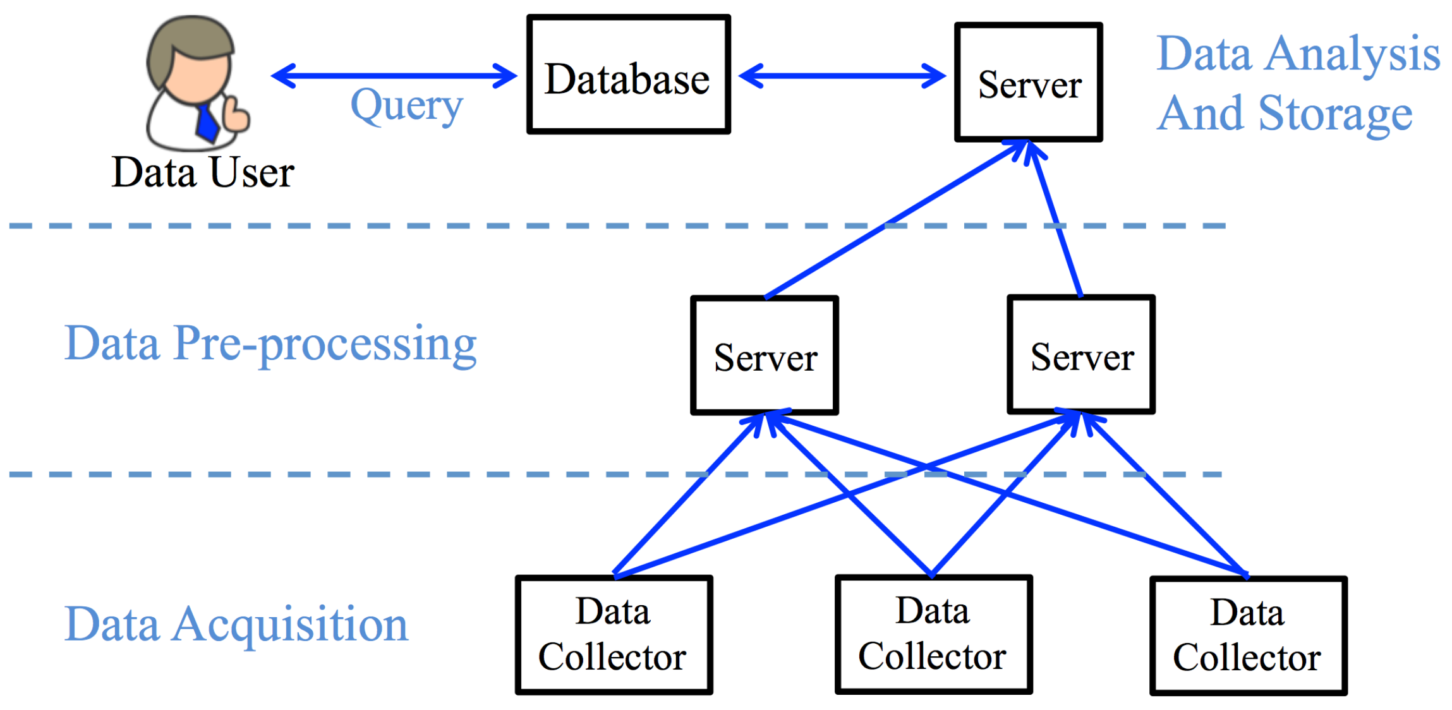
SWITCH

Background

This poster presents a network-transparent flexible provisioning agent for complex virtual infrastructure distributed over multiple data centres or cloud providers. This work is conducted in the context of the EU H2020 project SWITCH as part of a dynamic real-time infrastructure planner for cloud applications. This technology can be used by system developers to customize and provision virtual infrastructure for applications with critical performance constraints that may influence how and where application components are hosted.

Research Problems

Quality critical applications such as disaster early warning, live event broadcasting or real-time business collaboration often require customized virtual infrastructure with tailored service level agreements (SLAs) when migrated into a cloud environment. A typical architecture for a big data quality critical application is shown in the figure below.



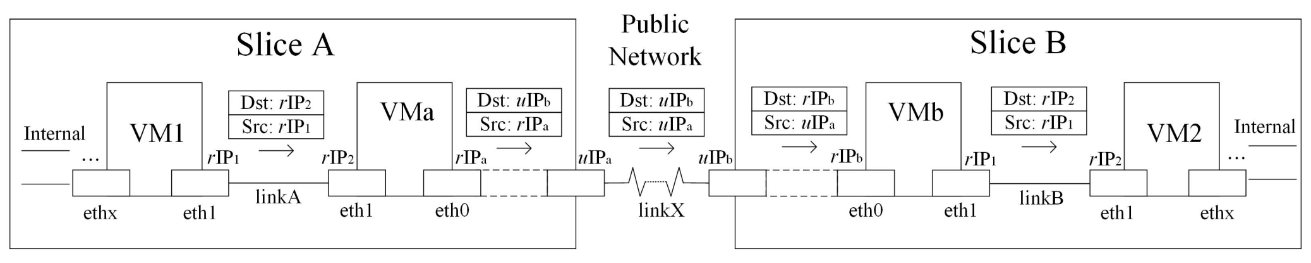
According to the current state of the Cloud in industry, we infer the following challenges and gaps when migrating this kind of quality critical application onto Cloud, focusing mainly on infrastructure provisioning.

- **Networked infrastructure**
  - The application workflow becomes more complex with a lot of components that need to communicate with each other. The virtual infrastructure must therefore realize a particular network topology.
  - Most current cloud providers cannot support this however; for example, Amazon EC2 can only allow users to describe private subnets, making it hard to build a complete topology.
- **Nearly real-time constraints**
  - Static constraints (e.g. network transmission time as data is processed), which restrict task scheduling before provisioning.
  - Runtime constraints (e.g. on failure recovery time), which affect how applications are managed at runtime, especially where the Cloud is not wholly reliable.
- **Geography**
  - In the figure above, not all the components of an application are on the Cloud.
  - Virtual infrastructure should be provisioned close to external components in order to satisfy any nearly real-time constraints on data delivery.
- **Auto-provisioning and federated cloud**
  - We need a way to provision the whole infrastructure and deploy applications automatically. Currently, some tools can only provision automatically at instance level, for example Chef.
  - The application may need more resources from other Clouds to provision a large scale infrastructure. It is a problem to combine these resources across multiple locales however.

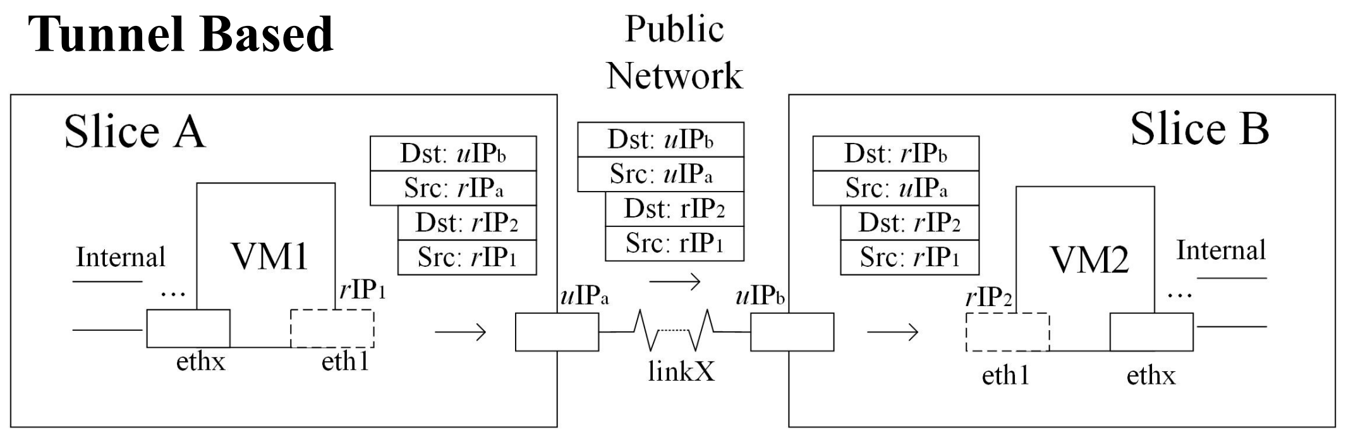
Approach

Connection Methods

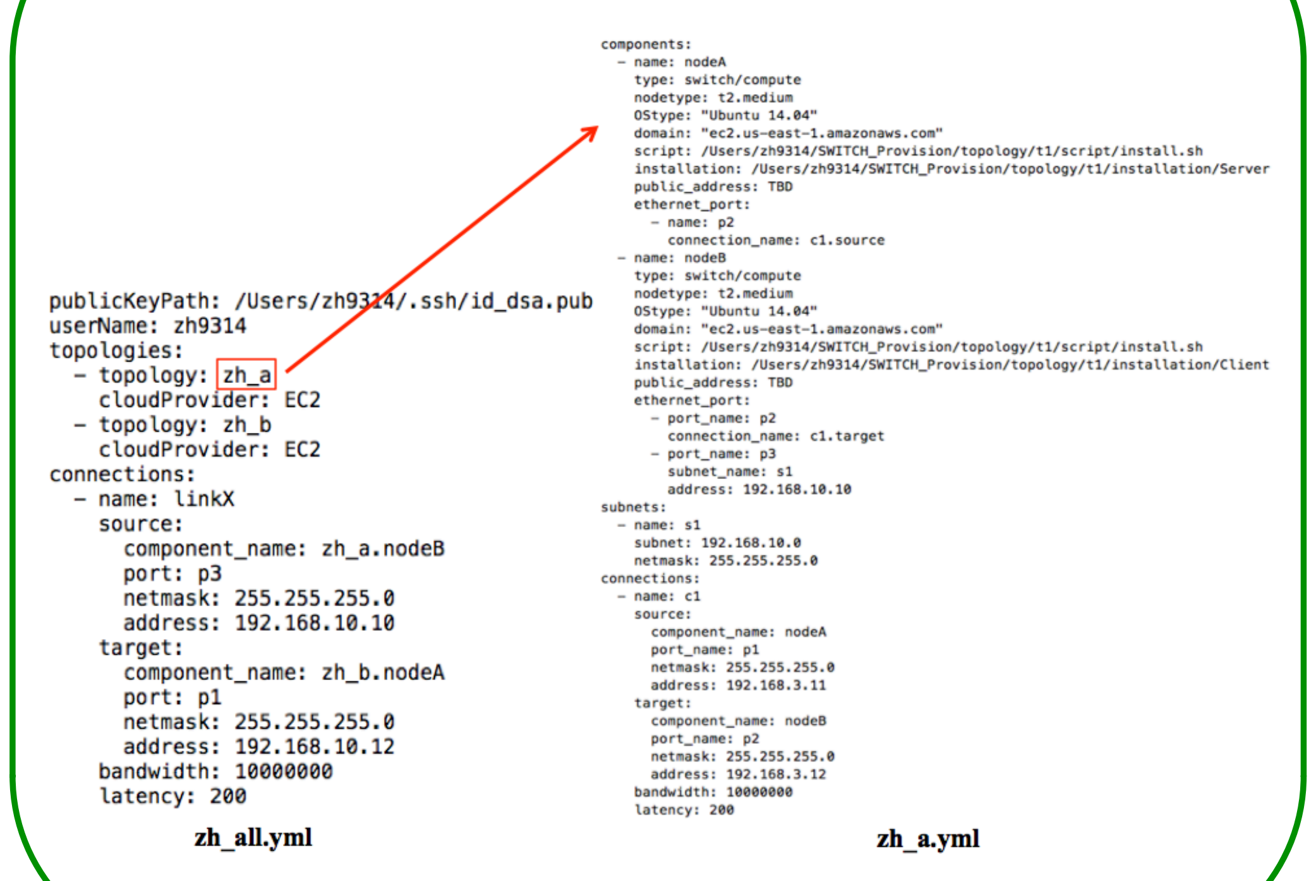
- **NAT Based**



- **Tunnel Based**



Standardized Multi-level Infrastructure Description

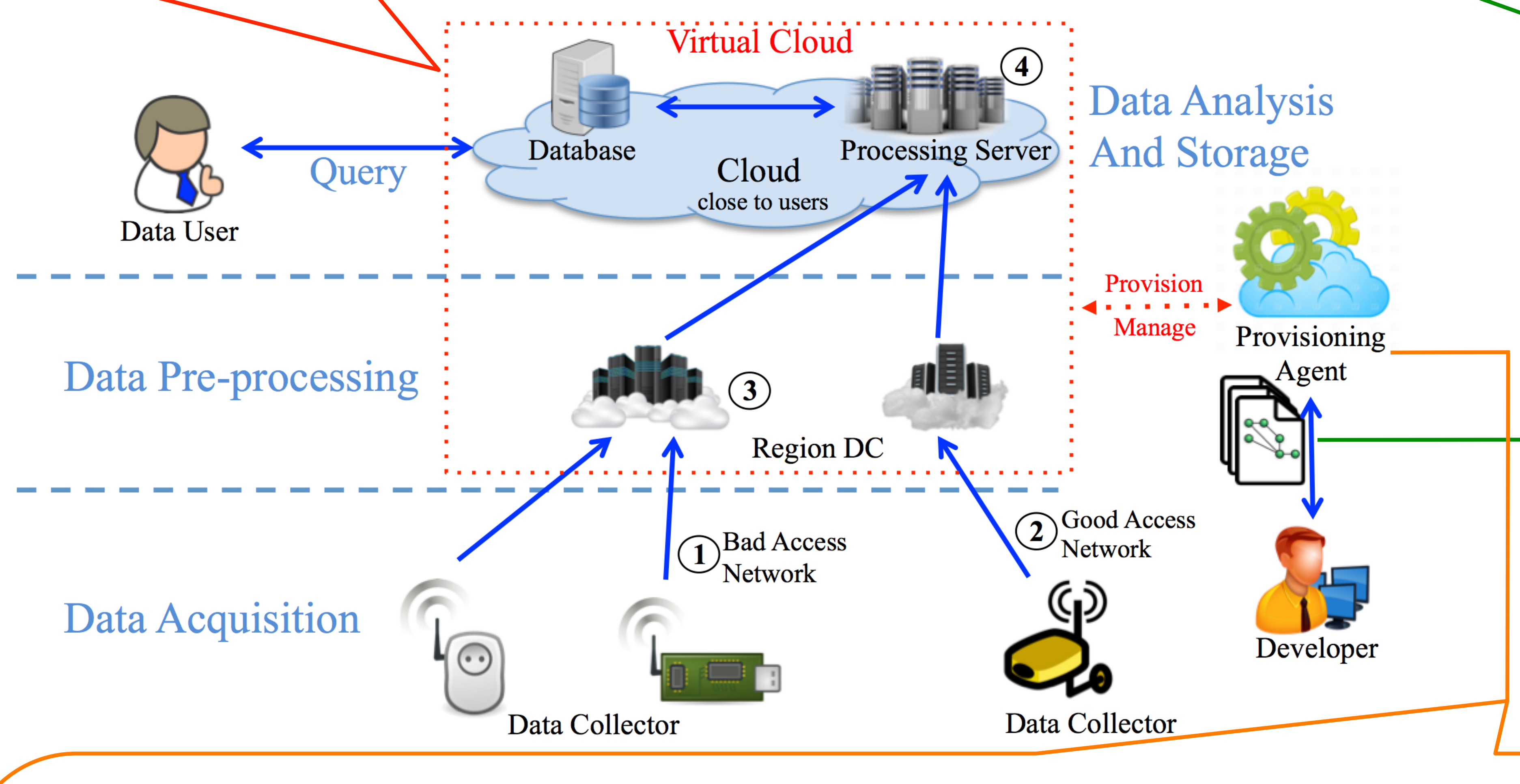


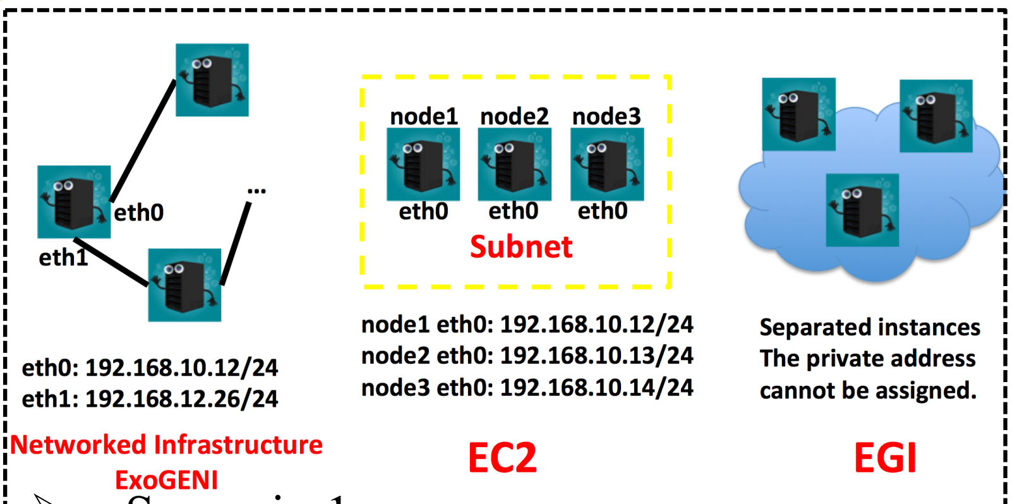
```
publicKeyPath: /Users/zh9314/.ssh/id_dsa.pub
userName: zh9314
topologies:
  - topology: zh_a
    cloudProvider: EC2
  - topology: zh_b
    cloudProvider: EC2
connections:
  - name: linkX
    source:
      component_name: zh_a.nodeB
      port: p3
      netmask: 255.255.255.0
      address: 192.168.10.10
    target:
      component_name: zh_b.nodeA
      port: p1
      netmask: 255.255.255.0
      address: 192.168.10.12
    bandwidth: 10000000
    latency: 200
```

zh_all.yml

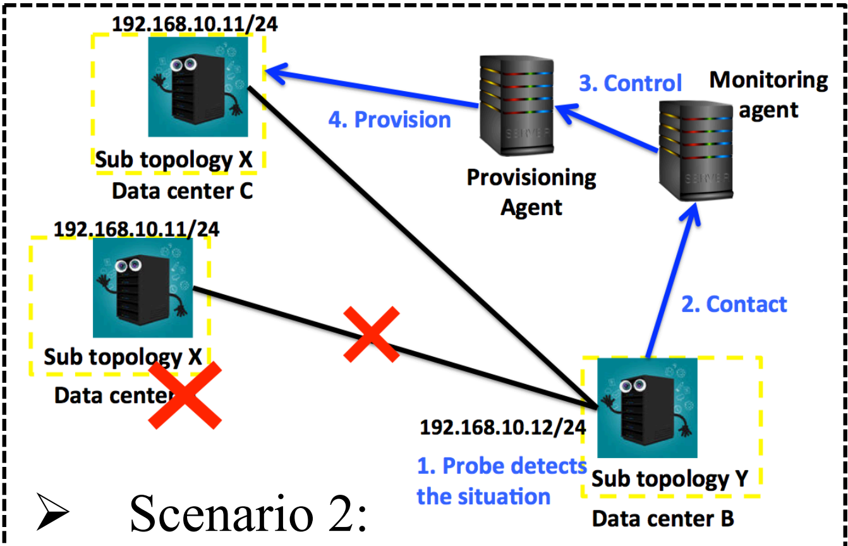
```
components:
  - name: nodeA
    type: switch/compute
    nodetype: t2.medium
    OStype: "Ubuntu 14.04"
    domain: "ec2.us-east-1.amazonaws.com"
    script: /Users/zh9314/SWITCH_Provision/topology/t1/script/install.sh
    installation: /Users/zh9314/SWITCH_Provision/topology/t1/installation/Server
    public_address: TBD
    ethernet_port:
      - name: p2
        connection_name: c1.source
  - name: nodeB
    type: switch/compute
    nodetype: t2.medium
    OStype: "Ubuntu 14.04"
    domain: "ec2.us-east-1.amazonaws.com"
    script: /Users/zh9314/SWITCH_Provision/topology/t1/script/install.sh
    installation: /Users/zh9314/SWITCH_Provision/topology/t1/installation/Client
    public_address: TBD
    ethernet_port:
      - port_name: p2
        connection_name: c1.target
      - port_name: p3
        subnet_name: s1
        address: 192.168.10.10
subnets:
  - name: s1
    subnet: 192.168.10.0
    netmask: 255.255.255.0
connections:
  - name: c1
    source:
      component_name: nodeA
      port_name: p1
      netmask: 255.255.255.0
      address: 192.168.3.11
    target:
      component_name: nodeB
      port_name: p2
      netmask: 255.255.255.0
      address: 192.168.3.12
    bandwidth: 10000000
    latency: 200
```

zh_a.yml

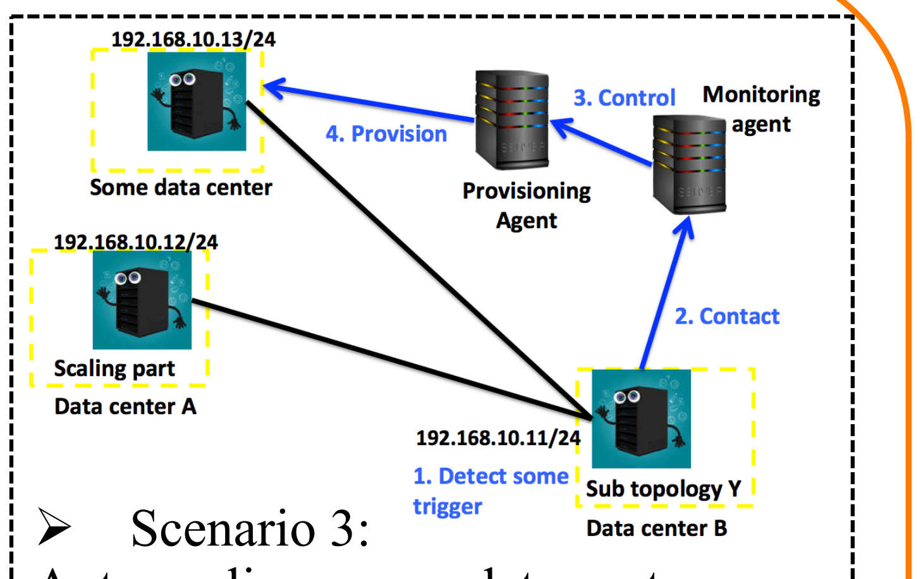




- Scenario 1: Infrastructure-level automatic provisioning of VMs and network links.



- Scenario 2: Fast recovery from some sudden failures.



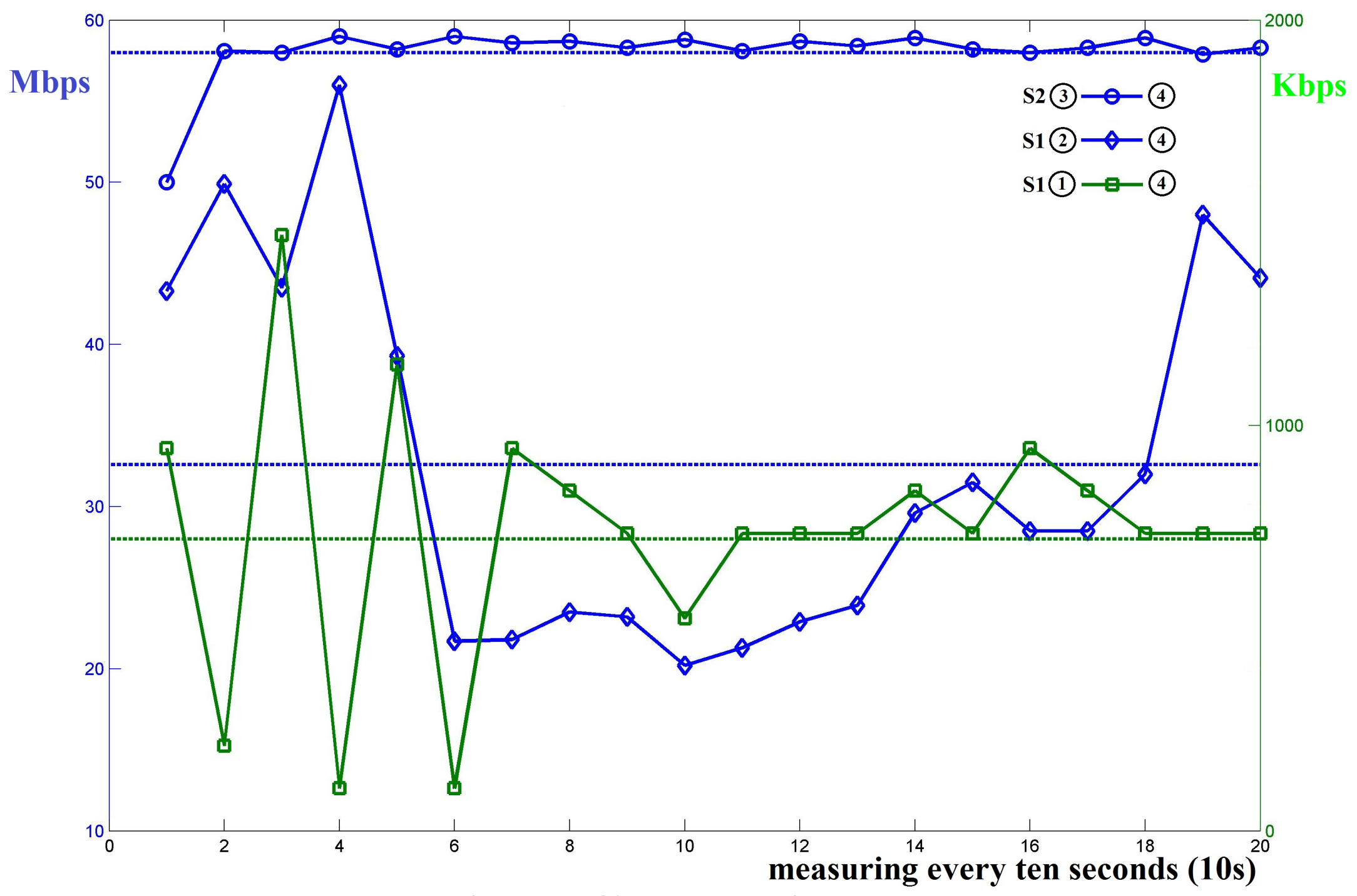
- Scenario 3: Auto-scaling across data centers or Clouds.

Application Scenarios

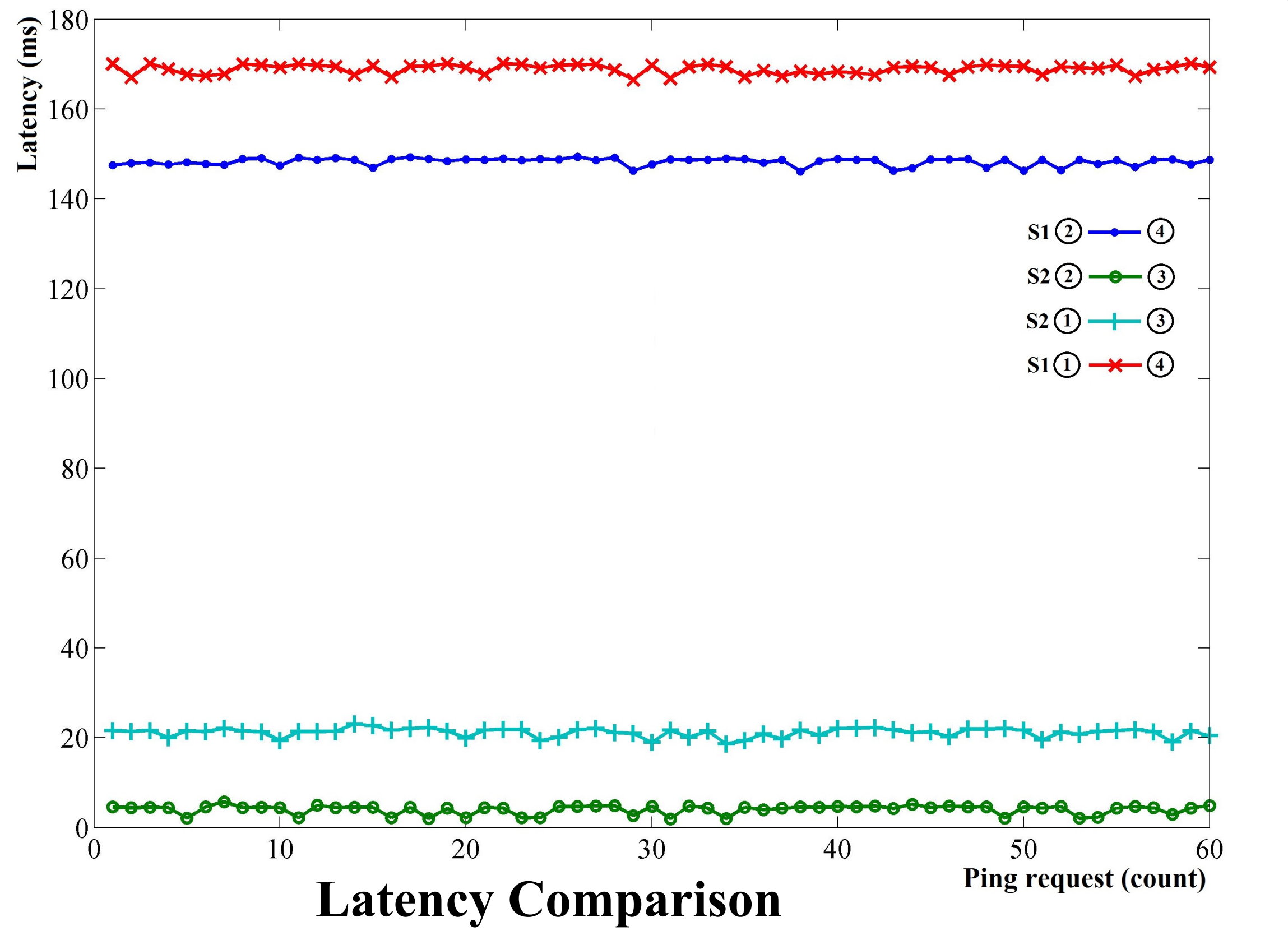
Our provisioning agent can distribute an application workflow across multiple locales while preserving network inter-connectivity between tasks. This is done via a choice of methods, and requires no special effort by the application developer or cloud provider.

Experiment Results

We have conducted experiments to study the impact of our solution architecture on application performance, testing the influence of task placement on latency and bandwidth.



Bandwidth Comparison



Latency Comparison

There are two main scenarios we need to compare:

- Scenario "S1" in the figure, is the deployment of all the components in one data centre without use of our agent.
- Scenario "S2" in the figure, is to adopt our solution, which is to distribute the components among multiple locales with our agent.

Table: Properties of objects used to test performance of linked tasks hosted in different locales.

Object Number	Subject	Computing Properties			Access Network Properties			Geography Properties	
		CPU Core	OS	Memory	Mode	Upload	Download	Cloud Provider	Location
①	Laptop	1 GB	MacOS	8 GB	WIFI(Home Network)	0.94 Mbps	8.59 Mbps	-	Amsterdam
②	Laptop	1 GB	MacOS	8 GB	WIFI(Eduroam)	193 Mbps	305 Mbps	-	University of Amsterdam
③	Virtual Machine	1 GB	Ubuntu 14.04	3 GB	Ethernet	-	-	ExoGENI	University of Amsterdam
④	Virtual Machine	1 GB	Ubuntu 14.04	3 GB	Ethernet	-	-	ExoGENI	CA, USA

Contact: Huan Zhou (h.zhou@uva.nl) Zhiming Zhao (z.zhao@uva.nl)
SWITCH grant no. 643963
Coordinator: dr. Zhiming Zhao, prof. dr. Cees de Laat
SWITCH online: http://www.switch-project.eu, or twitter @SWITCHprojectEU.

Universiteit van Amsterdam
Wellness Telecom
Cardiff University
Univerza v Ljubljani
Beia Consult International
MOG
COMMIT/