@

as a Research topic

Beyond the clouds - The Discovery initiative

Adrien Lebre / Matthieu Simonin / Ronan-Alexandre Cherrueau

orange™

Who are We?

Adrien Lebre

Inria Researcher

Discovery Chair
http://beyondtheclouds.github.io

Fog/Edge/Massively Distributed WG Chair
https://wiki.openstack.org/wiki/Fog_Edge_Massively_Distributed_Clouds

Matthieu Simonin

Inria Research Engineer
Discovery Technical Architect

Fog/Edge/Massively Distributed WG and Performance team Contributor

Ronan-Alexandre Cherrueau

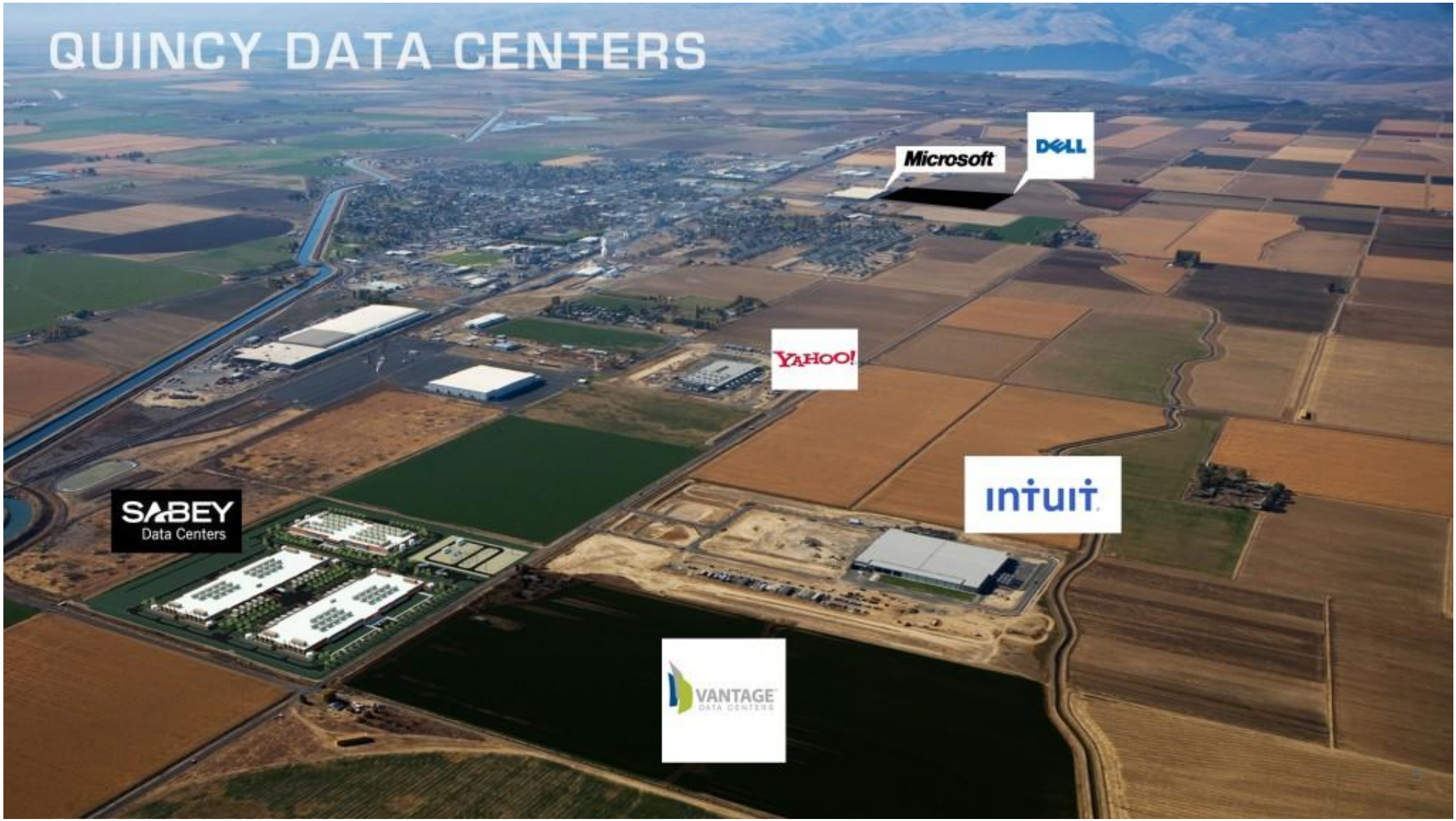
Discovery Initiative Researcher Engineer

Fog/Edge/Massively Distributed WG and Performance team Contributor

EnOS main developer
http://enos.readthedocs.io

From mainframes to...


From mainframes to...
...“larger” mainframes


QUINCY DATA CENTERS
Microsoft
DELL
YAHOO!
intuit
SABEY
Data Centers
VANTAGE
DATA CENTERS

2012 - 2013
Major brakes for the adoption of the CC model

Jurisdiction concerns
Reliability

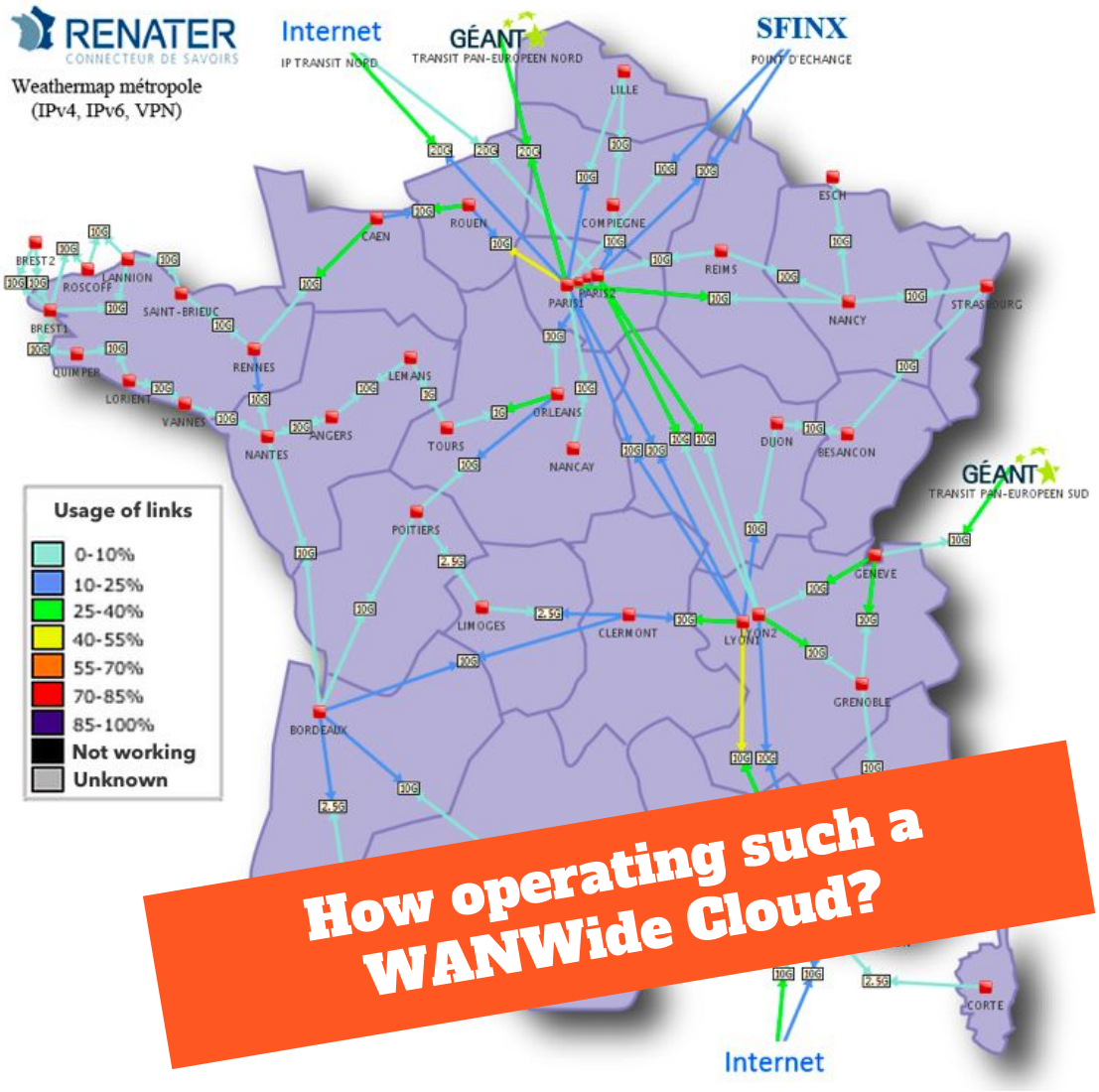
CC Distance (network overheads)

2015 -
NFV / SDN / IoT

Discovery Vision

Bring Clouds back to the cloud

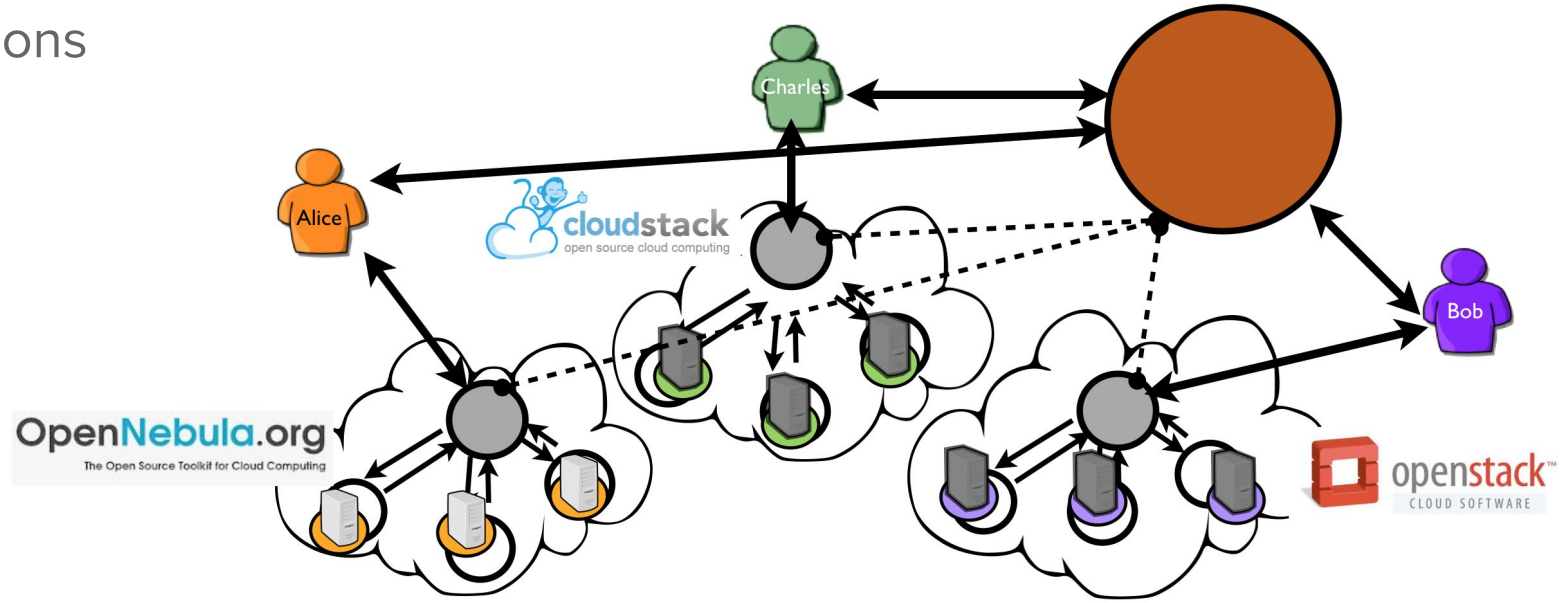
- Leverage the concept of µDC/nDC to extend any point of presence of network backbones (aka PoP) with servers
- From network hubs up to major DSLAMs that are operated by telecom companies, network institutions…



The Broker Approach

Sporadic (hybrid computing/cloud bursting) almost ready for production

Brokers are rather limited to simple usages and not advanced administration operations



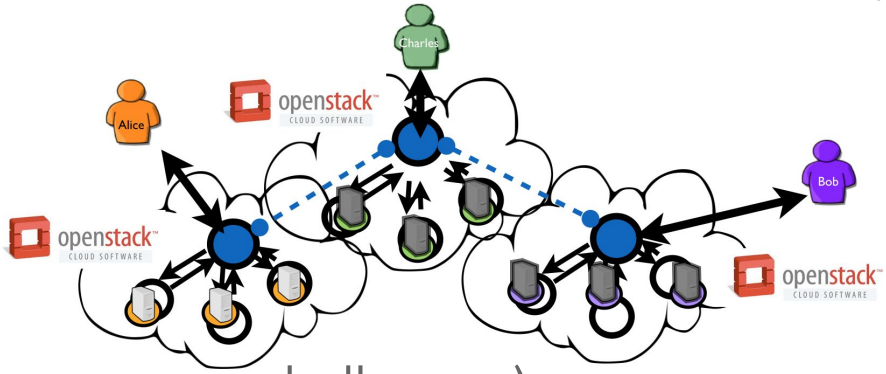
Advanced brokers must reimplement standard IaaS mechanisms while facing the API limitation

Would OpenStack be the Solution?

Do not reinvent the wheel... it's too late
OpenStack (20Millions of LOC, 3M just for the core-services)

Discovery objectives (overview)

- Study to what extent the current OpenStack mechanisms can handle such massively distributed infrastructures
- Propose revisions/extensions of internal mechanisms when appropriate

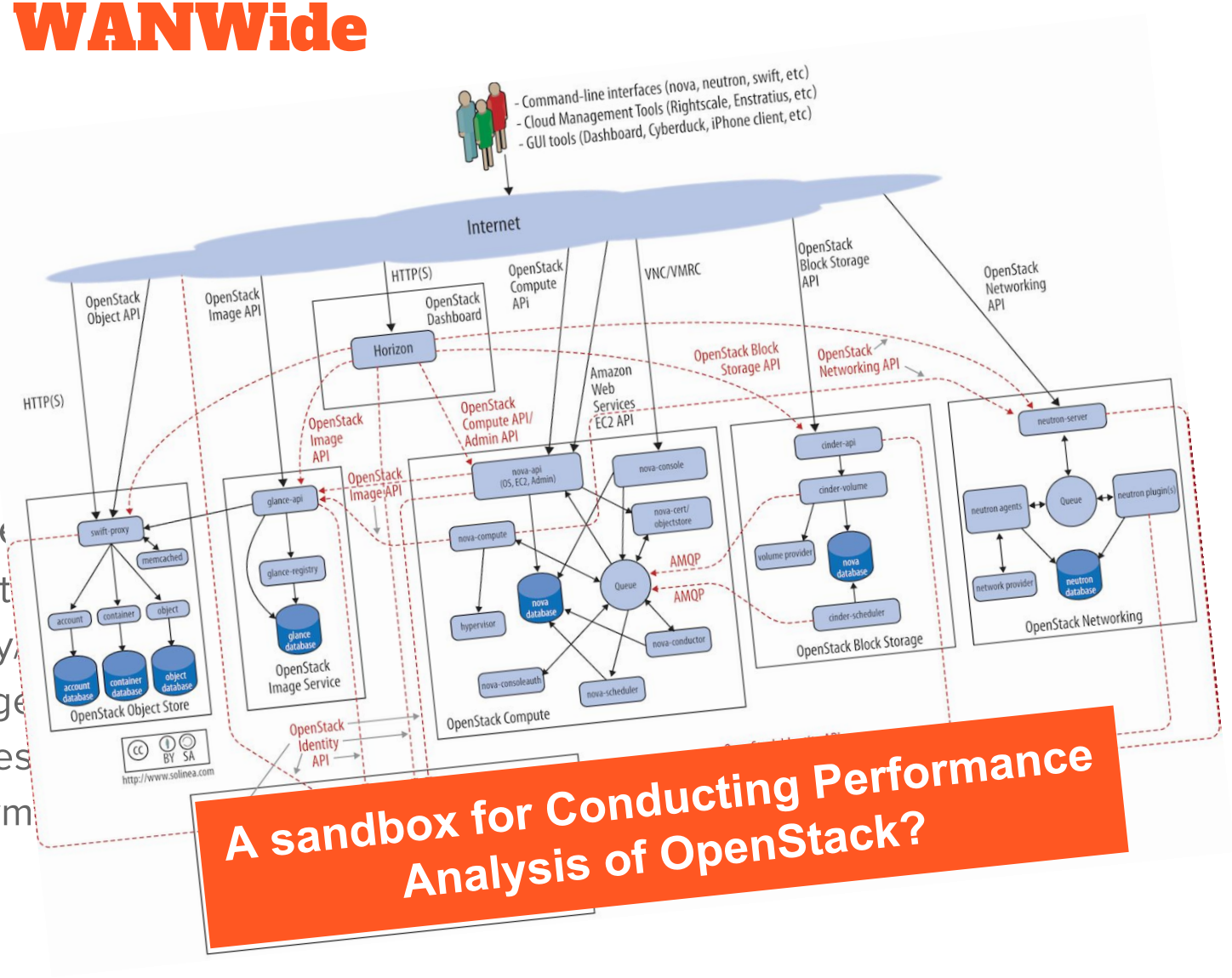


From SQL to NoSQL backend...
(a research PoC, just the top of the iceberg, numerous challenges)

Toward a Holistic Framework for Conducting Scientific Evaluations of OpenStack
EnOS, A tool for diving into OpenStack and performing scientific investigations

OpenStack WANWide

- Several deplo
  - control ser
    nodes rem
  - Segregatio
    - cells (n
    - regions
- Which one is th
  - No real funct
    - Latency
    - Message
    - Changes
    - Deploym

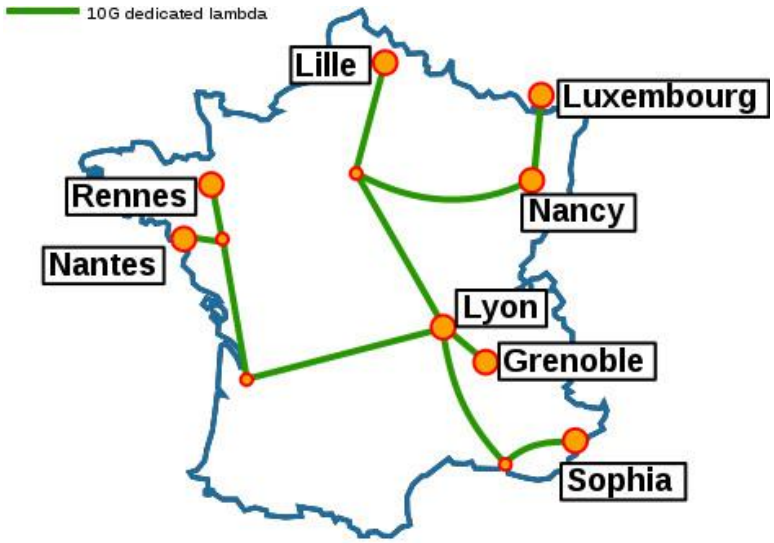


A sandbox for Conducting Performance Analysis of OpenStack?

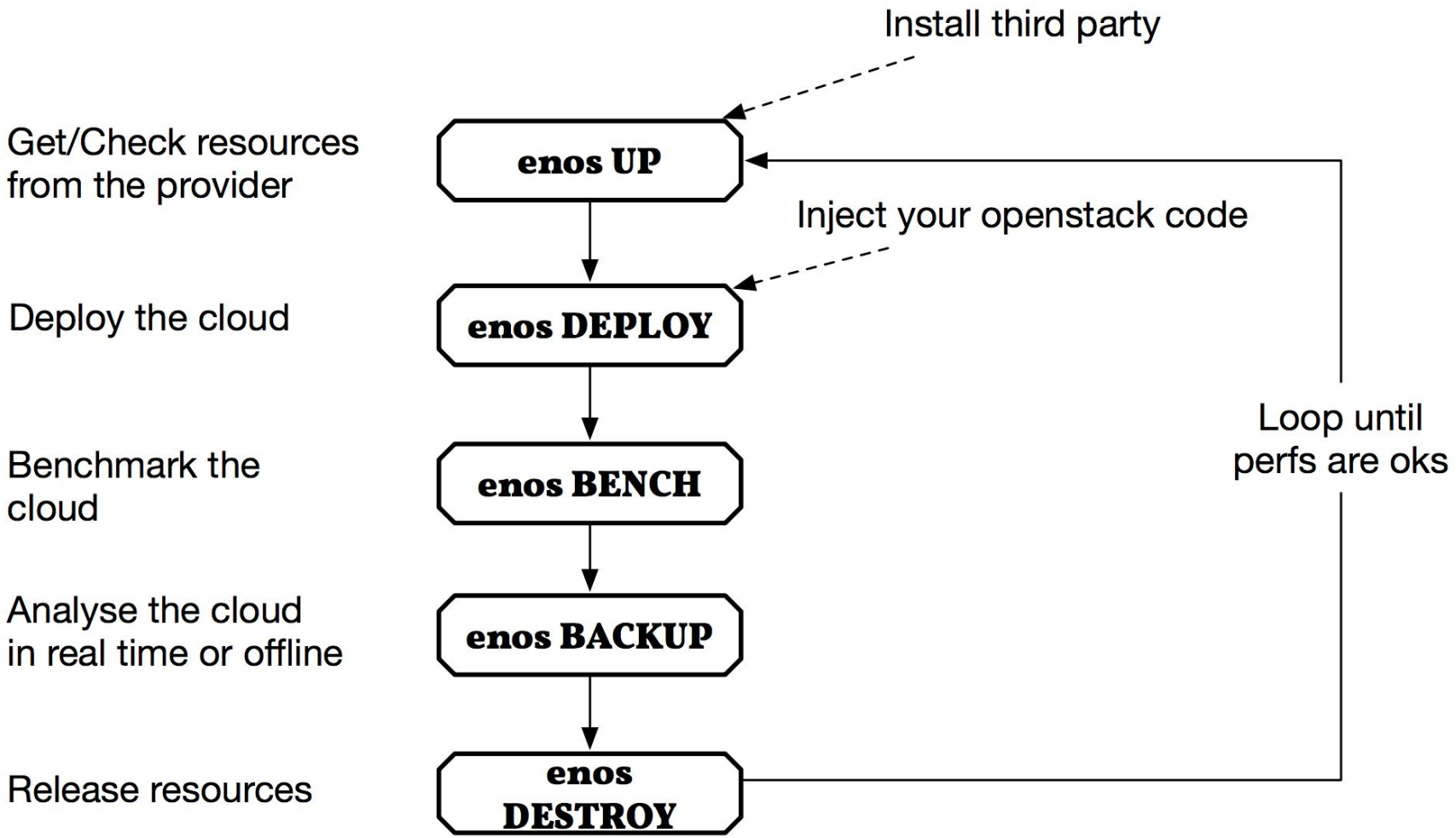
EnOS: Experimental Env. for OpenStack

- Background
  - Grid'5000
  - Reproducible Research
  - Evaluation of complex systems (at scale)

- Motivation:
  - Flexible performance study :
    - At small and large-scale / multi-site
    - Under different network topologies
  - Between different releases
  - With different kind of benchmarks



EnOS: Workflow



EnOS: Experimental Env. for OpenStack

- https://github.com/beyondtheclouds/enos
- Implemented on top of Kolla (Ansible + Docker deployment)
  - leverage everything's possible by Kolla (upstream what is not yet possible)
- Supports different providers
  - Vagrant (local machine)
  - Grid'5000
  - Openstack
  - Chameleon (Openstack with ironic)
- Benchmarks : Rally, Shaker. OsProfiler support.
- Available on pypi under GPLv3

EnOS deploy – Resource/Topology Description

```
$ cat ./basic.yml
resources:
  clusterA:
    control: 1
    network: 1
  clusterB:
    compute: 50

$ enos deploy -f basic.yml
```

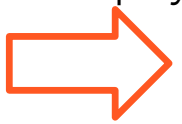
```
$ cat ./advanced.yml
resources:
  clusterA:
    control: 1
    network: 1
    nova-conductor: 5
  clusterB:
    compute: 50

$ enos deploy -f advanced.yml
```

```
$ cat ./network-topo.yml
resources:
  grp1:
    clusterA:
      control: 1
      network: 1
      nova-conductor: 5
  grp2:
    clusterB:
      compute: 50

network_constraints:
  - src: grp1
    dst: grp2
    delay: 100ms
    rate: 10Gbit
    loss: 0%
    symetric: true

$ enos deploy -f network-topo.yml
```

EnOS deploy – Under the Hood

```
resources:
  grp1:
    clusterA:
      control: 1
      network: 1
  grp2:
    clusterB:
      compute: 50

network_constraints:
  delay: 100ms
  rate: 10Gbit
  loss: 0%
```

```
$ enos deploy
```



1. Provider gets 2 nodes on `clusterA`, 50 nodes on `clusterB` and returns node's IP addresses
2. EnOS provisions nodes with Docker daemon
3. EnOS installs OpenStack using Kolla
4. EnOS sets up bare necessities (flavors, cirros image, router, ...)
5. EnOS applies network constraints between `grp1` and `grp2` using tc

---

- Provider to get testbed resources
  - Resources: anything running a Docker daemon and EnOS can SSH to + some IPs
  - Existing Provider: Vagrant (VBox/Libvirt), Grid'5000, Chameleon, OpenStack
  - ~500 LoC each
- Kolla to deploy OpenStack over testbed resources
- TC to apply network constraints

EnOS bench

- Benchmarks description

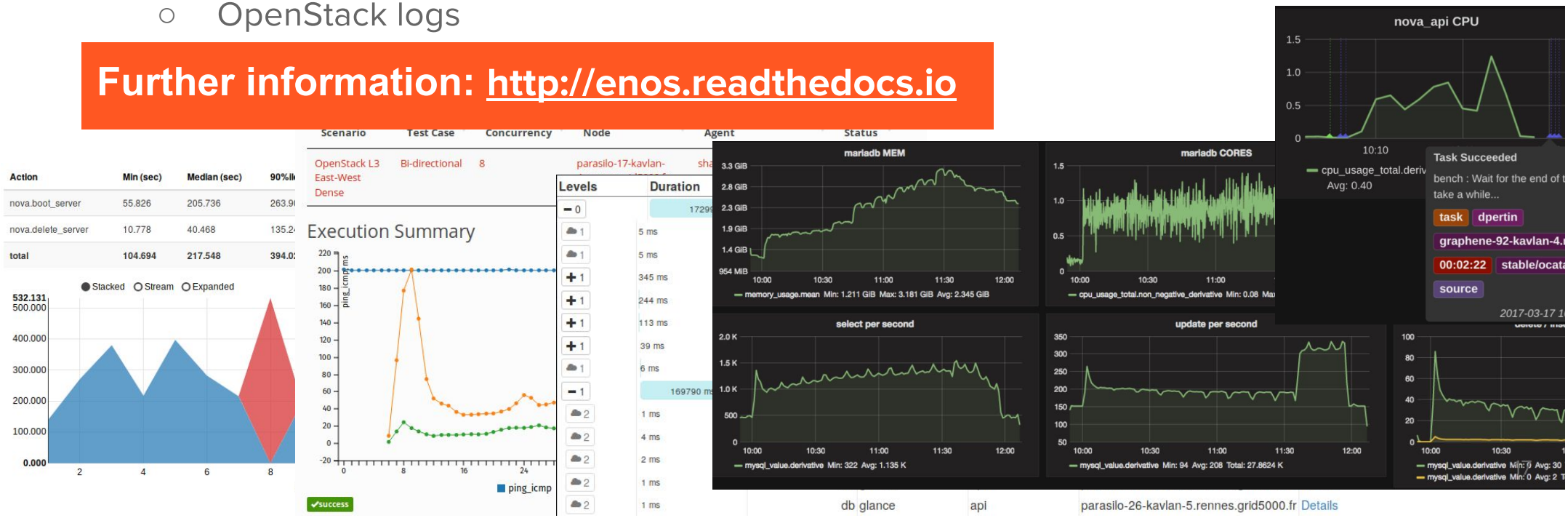
```
$ cat ./run.yml

rally:
  args:
    concurrency: 5
    times: 100
  scenarios:
    - name: boot and list servers
      file: nova-boot-list-cc.yml
      osprofiler: true
    - …
shaker: ...

$ enos bench --workload=run.yml
```

- Under the hood
  - Rally: control plane benchmark
  - Shaker: data plane benchmark
  - OSProfiler: code profiling
  - Monitoring stack: collect resources consumption per service/node/cluster

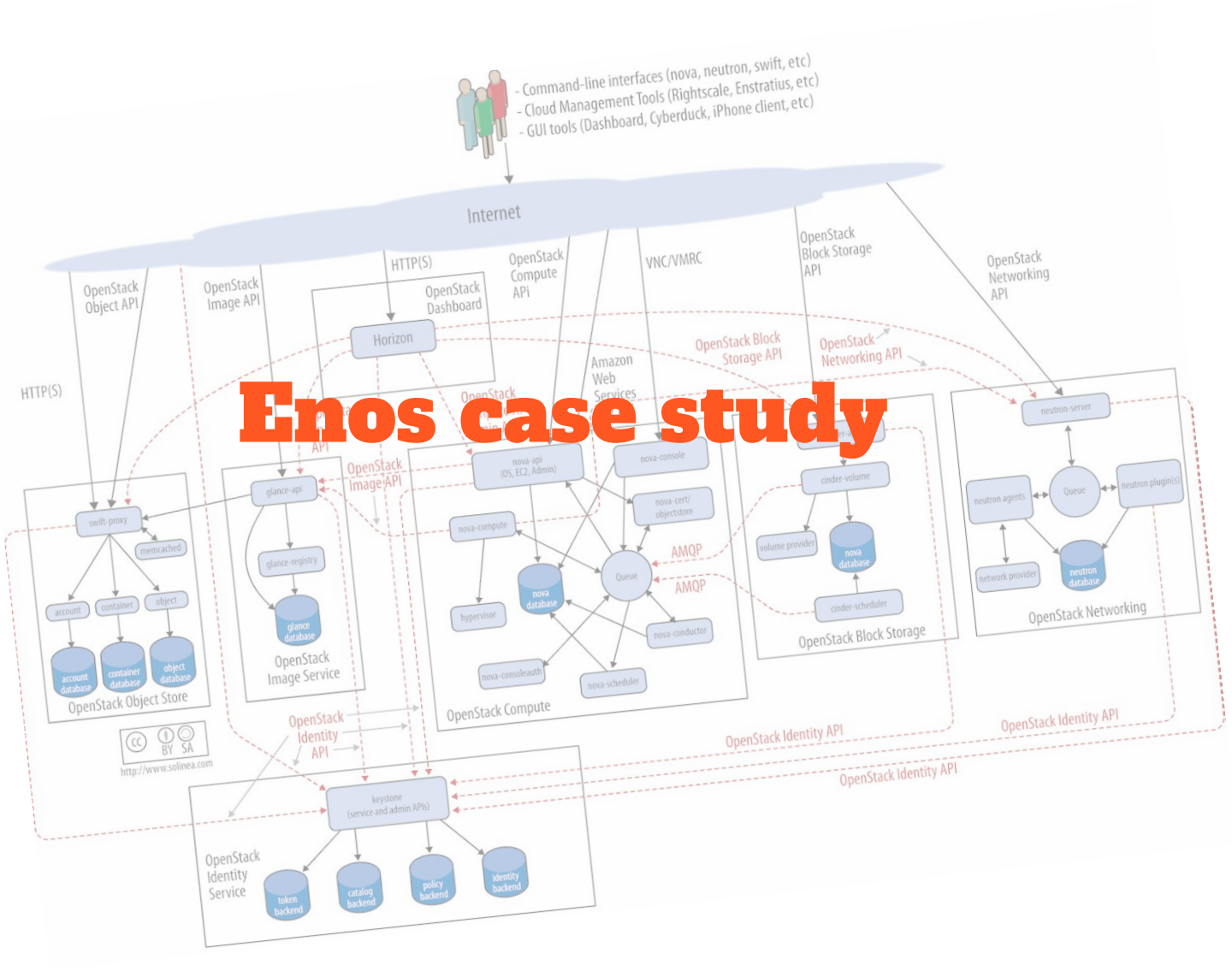
EnOS backup

- `enos backup` produces a tarball with
  - Rally/Shaker reports
  - OSProfiler traces
  - InfluxDB database with recorded time series
  - OpenStack logs

Further information: http://enos.readthedocs.io

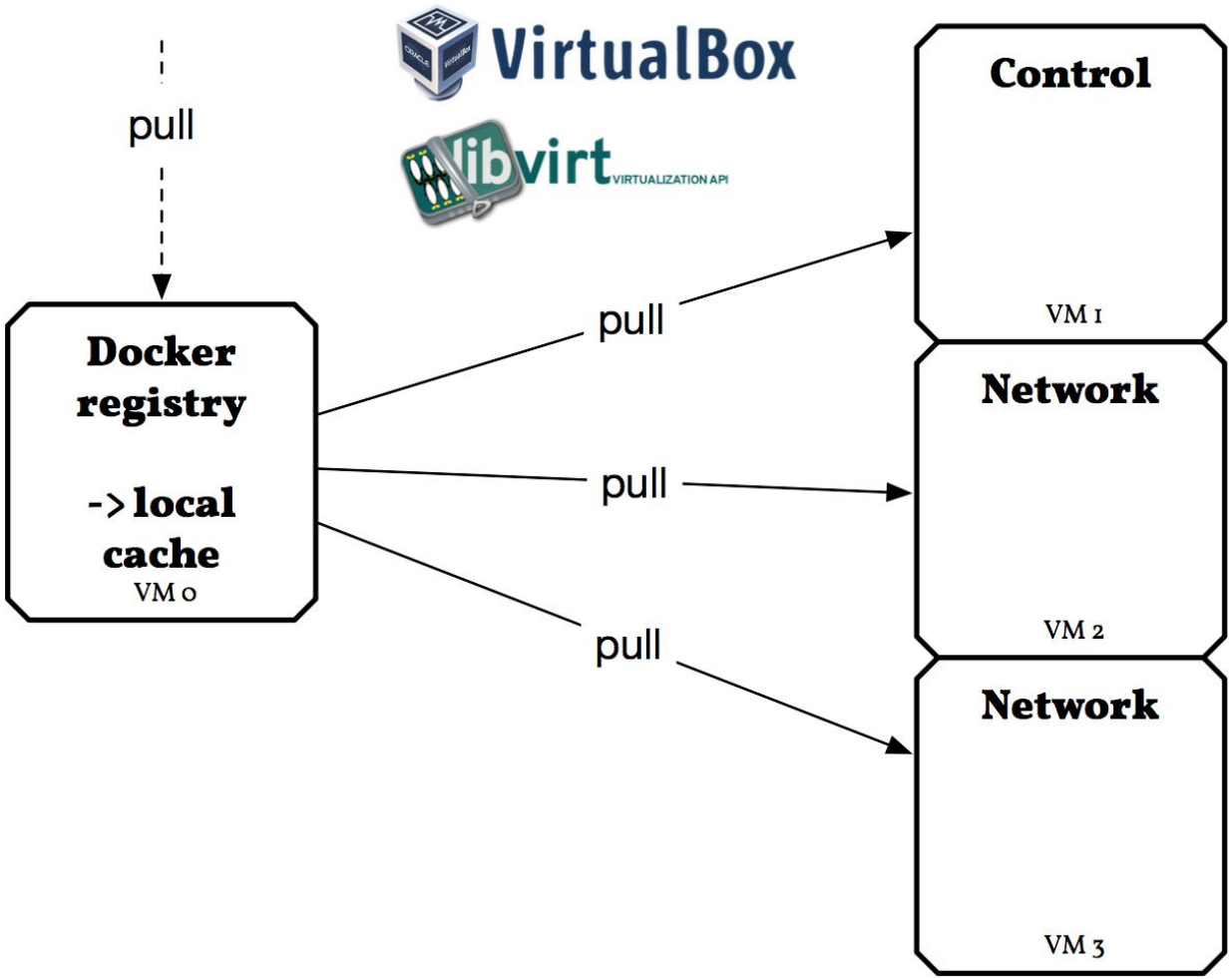


Enos case study



Local Test/Development environment

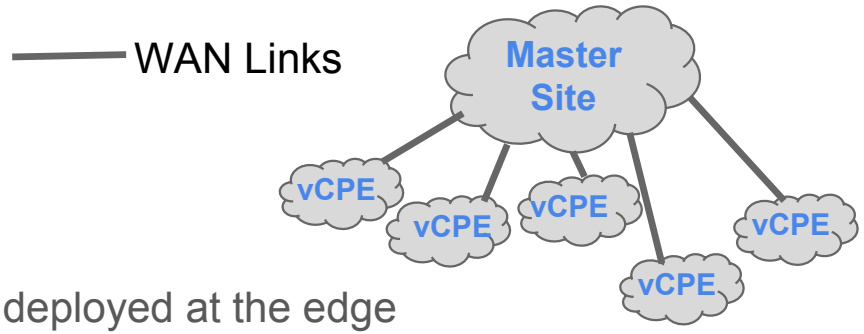
- fast iteration on the deployment
  - EnOS dev.
  - Kolla dev.
  - Configuration validation



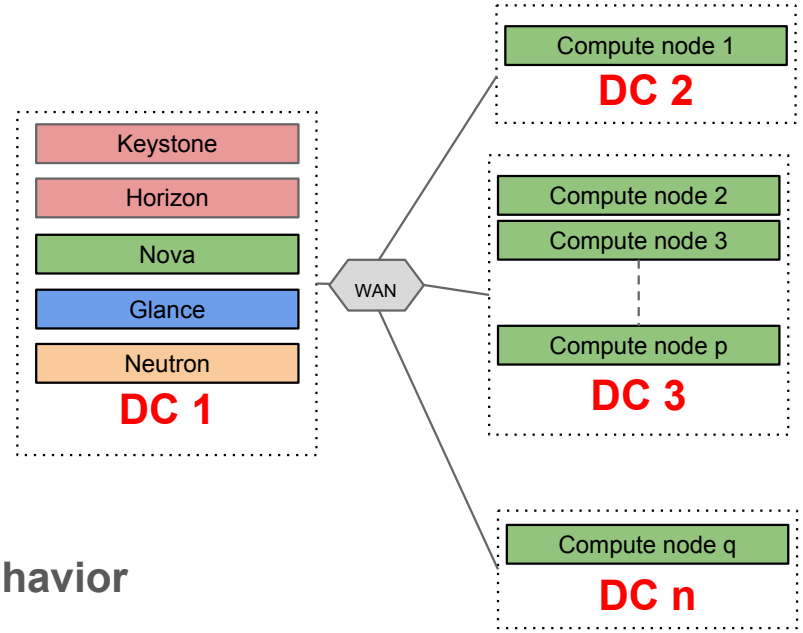
Large scale deployment

- Chasing 1000 nodes scalability (Barcelona summit)
  - Joint work with Mirantis
  - Experimentation made on Grid'5000
  - Performance metrics / bottlenecks of OS
  - Lot of interests - results on the performance working group wiki.

OpenStack WANWide



- A Single OpenStack to operate remote compute resources deployed at the edge
  - All control services are deployed into the master.
  - The RabbitMQ bus is deployed across all locations (i.e., through each server composing the infrastructure)
- Pros: simple
- Cons:
  - security management for RPC message and port, Single Point of Failure…
  - Scalability (not addressed in this presentation, see “Chasing 1000 Nodes Scale”, Barcelona Summit 2016)
  - **Network latency/throughput impacts on functional behavior and performance degradations**.

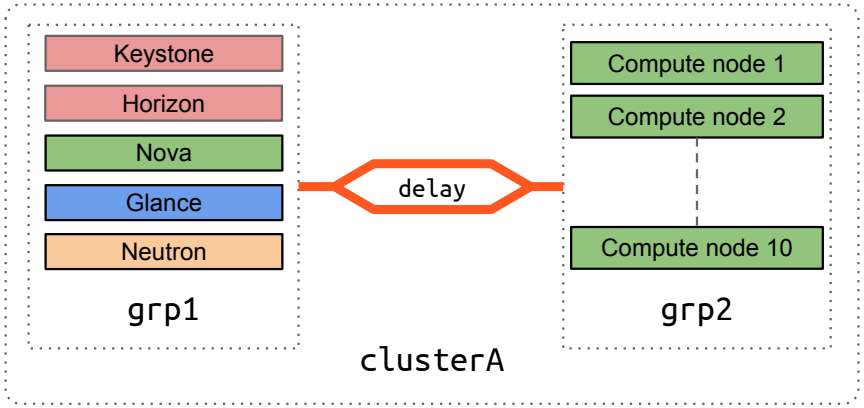


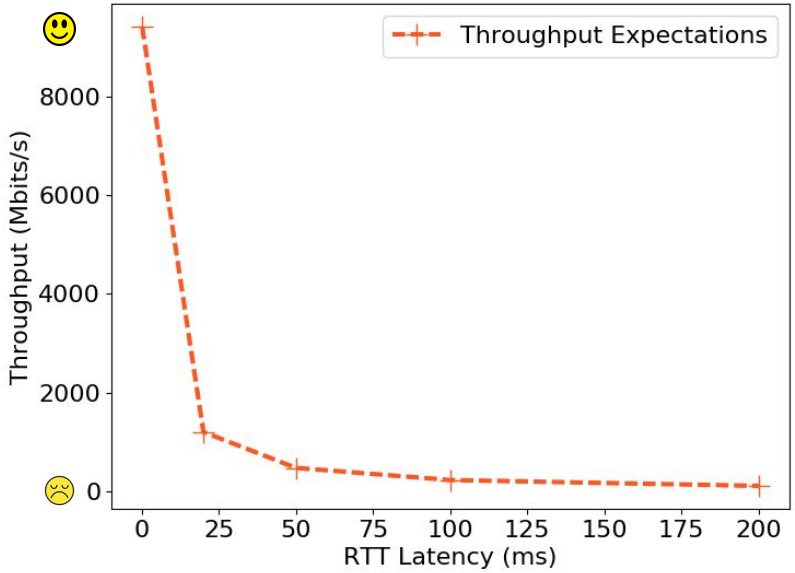
TestBeds Grid'5000 + 

- Experiments runs independently on both testbeds in a fully automatized manner (Software defined experiments leveraging EnOS)
- 250 benchmarks (approx. 100 running hours) on each testbed.
- Results lead to the same conclusion whatever the testbed (collected performance are almost identical).
- Experimental setup: https://github.com/BeyondTheClouds/enos-scenarios/
- Results: http://enos.irisa.fr/html/

Latency Impact (Experiment #1)

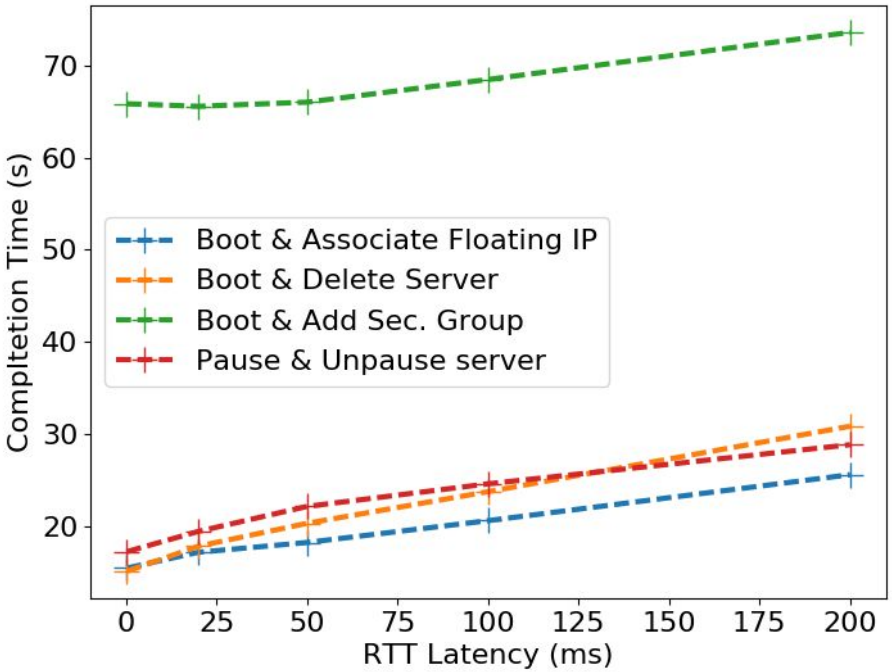
```
$ cat ./wan-exp1.yml

resources:
  grp1:
    clusterA:
      control: 1
  grp2:
    clusterA:
      compute: 10

network_constraints:
  delay: 0ms # 10ms, 25ms, 50ms, 100ms
  loss: 0%
  rate: 10Gbit
  src: grp1
  dest: grp2
  symetric: true

$ enos deploy -f wan-exp1.yml
```



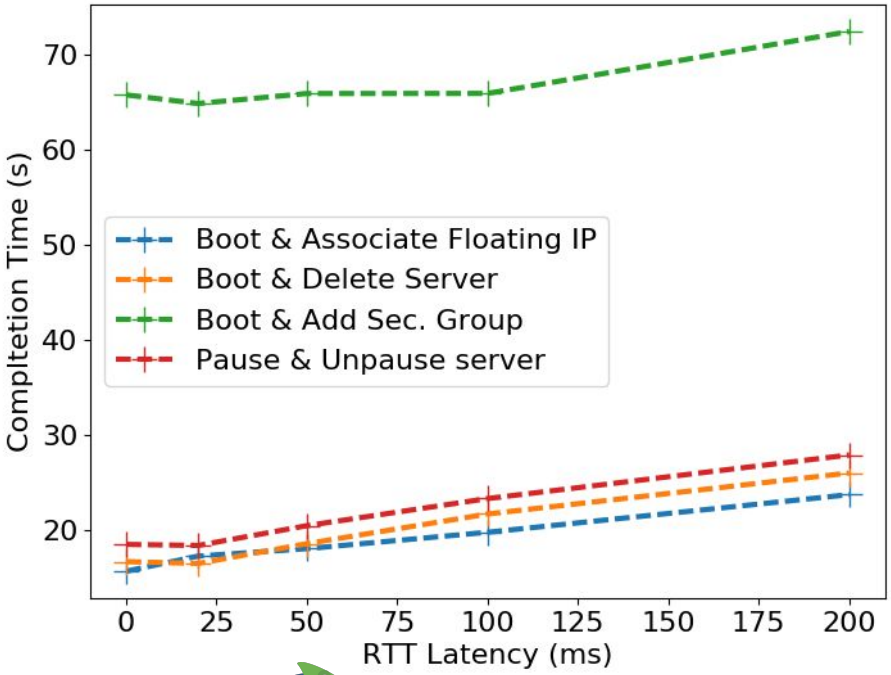


Find the scenario at https://github.com/BeyondTheClouds/enos-scenarios/tree/master/wan/cpt10

Latency Impact – Control Plane (Rally Metrics)



Grid'5000

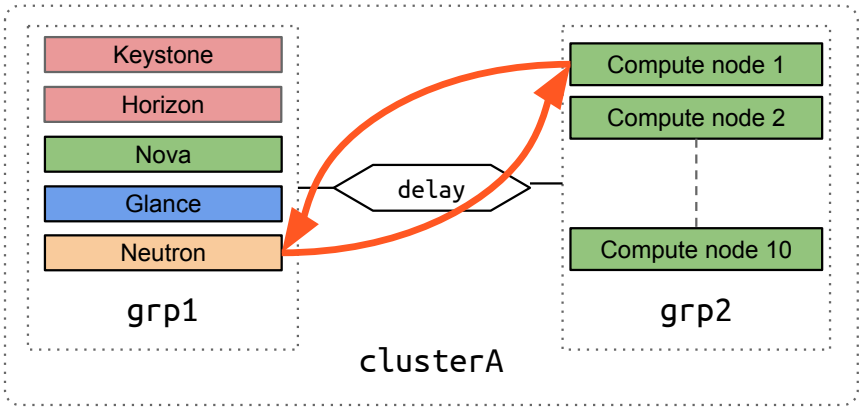


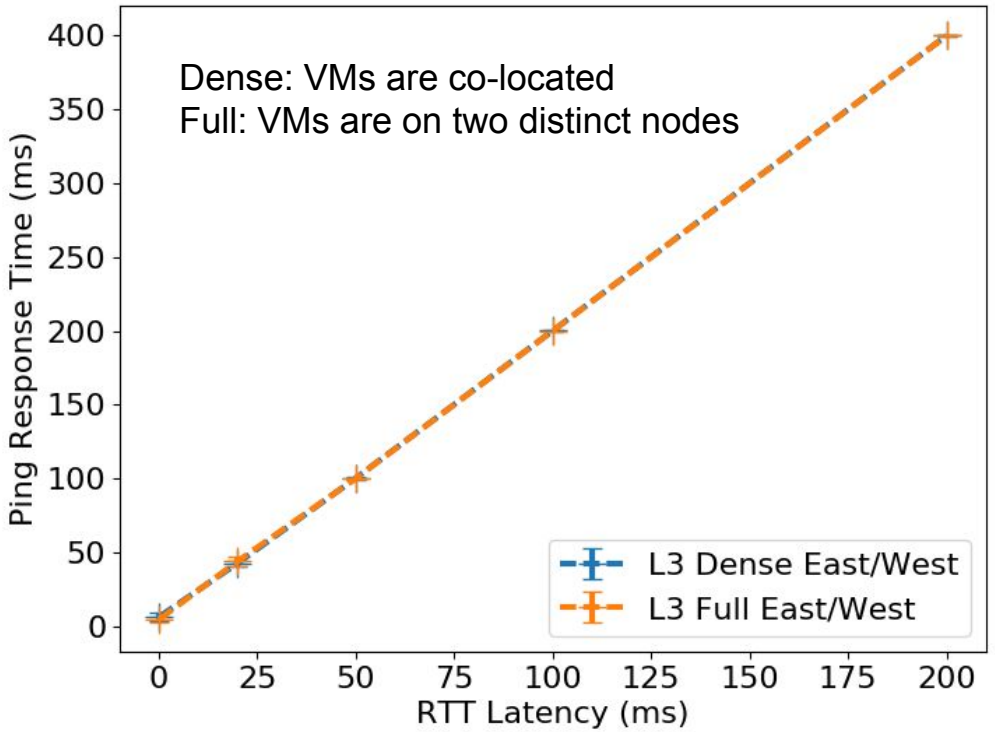
Chameleon

Trends are similar (which is what we expected !)

Latency Impact – Data Plane (Shaker Metrics)

```
$ cat ./run.yml

rally: ...
shaker:
  - file: openstack/dense_l3_east_west.yml
  - file: openstack/full_l3_east_west.yml

$ enos bench --workload=run.yml
```

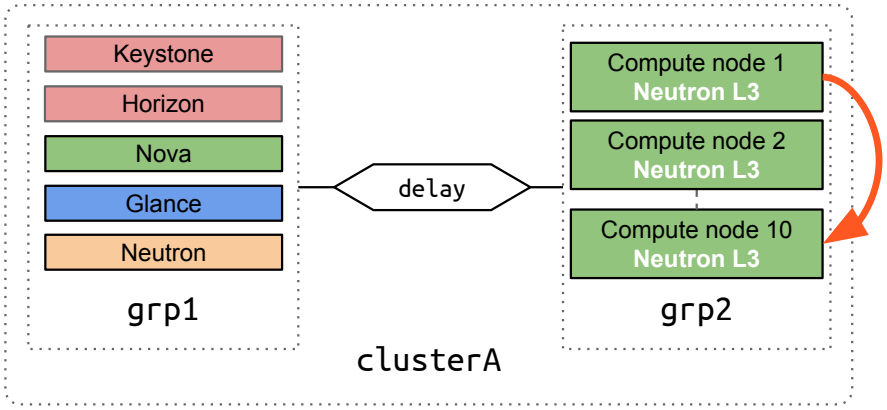




- Ping response time is twice the RTT (which corresponds to the normal workflow)

Latency Impact with DVR (Experiment #2)

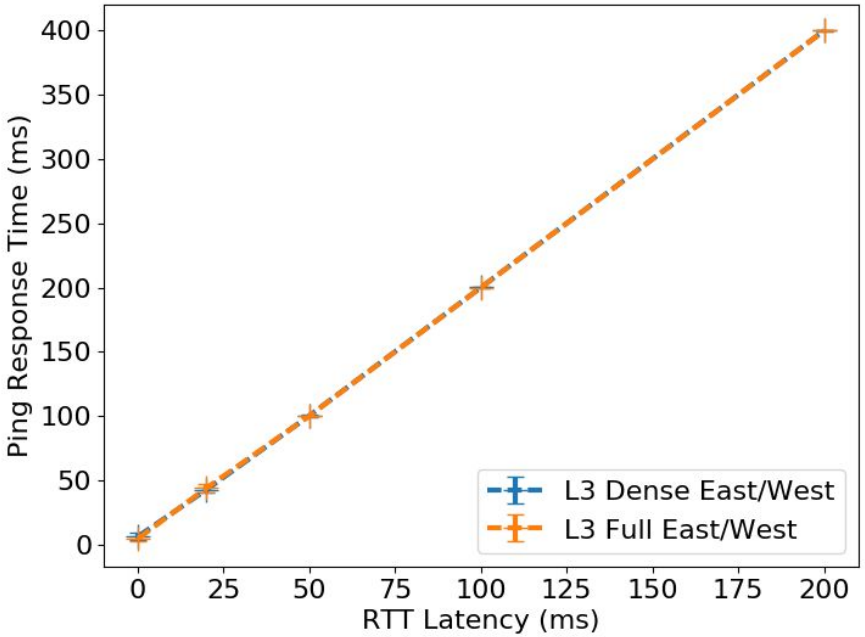
- You say DVR?
  - Distributed Virtual Routing
  - inter-tenant network is distributed to the compute nodes



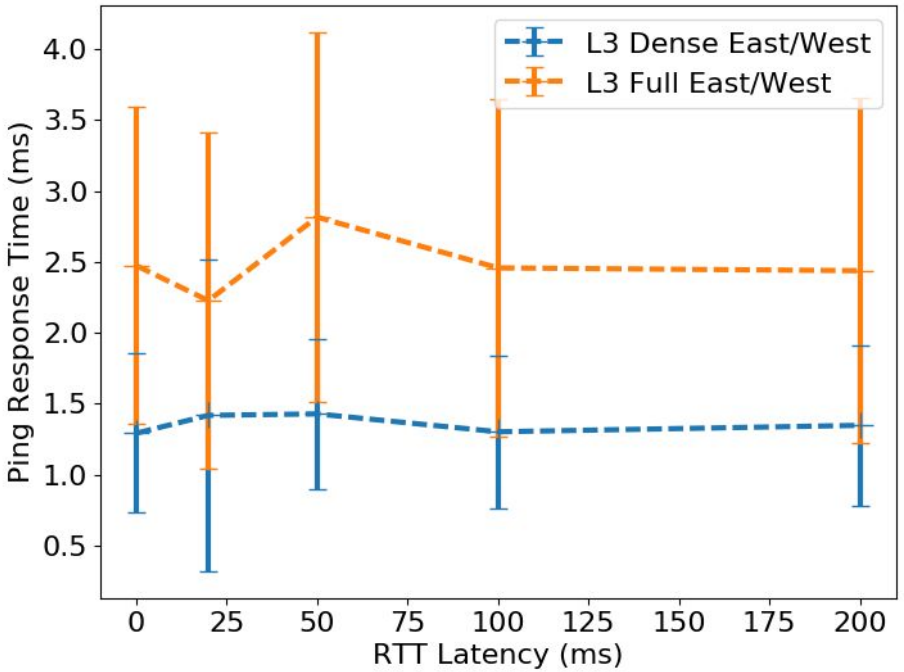

```
$ cat ./wan-exp2.yml

resources: ...

network_constraints: ...

kolla:
  enable_neutron_dvr: true

$ enos deploy -f wan-exp2.yml
```

Find the scenario at https://github.com/BeyondTheClouds/enos-scenarios/tree/master/wan/cpt10-dvr

Latency Impact with DVR – Data Plane



Without DVR
(2*RTT)



With DVR
(LAN RTT)

Critical change in WAN context

Find the scenario at https://github.com/BeyondTheClouds/enos-scenarios/tree/master/wan/cpt10-dvr

Conclusion

- **EnOS:** Experimental **environment for conducting OpenStack Performance Analyses**
  - **Evaluations are conducted on real deployments** (not by leveraging DevStack)

- **What's next:**
  - EnOS presentation/Tutorial at Sydney
  - On-going actions
    - Focus on **AMQP alternatives** (Apache Qpid Dispatch Router/ZeroMQ/...)
    - **From SQL to noSQL to newSQL (Cockroach DB as a backend for OpenStack services)**

- **Edge Computing, a key element for the OpenStack ecosystem** (opendev event, SF, Sept 2017)

  - **Join us - FEMDC team (IRC meeting on Wednesday, every two weeks)**

openstack @ Inria as a Research topic

Beyond the clouds - The Discovery initiative

Adrien Lebre / Matthieu Simonin / Ronan-Alexandre Cherrueau