Testing ARES on the GTS framework: lesson learned and open issues

Mauro Femminella
University of Perugia
mauro.femminella@unipg.it

Outline

- What is ARES
- What testing on GTS?
- Our solution
- Performance evaluation
- Conclusion and open issues

ARES: Advanced networking for the EU genomic Research

- Genomic processing is expected to become more demanding than top three applicative domains (YouTube, Twitter, astronomy)*:
  - data acquisition, storage, distribution, and analysis
- ARES is a networking & computing overlay to efficiently process genomic data sets, consisting of
  - Routers
  - Tenant in data centers

*Stephens ZD et al., "Big Data: Astronomical or Genomical?", PLoS Biol, 13(7): e1002195. doi:10.1371/journal.pbio.1002195

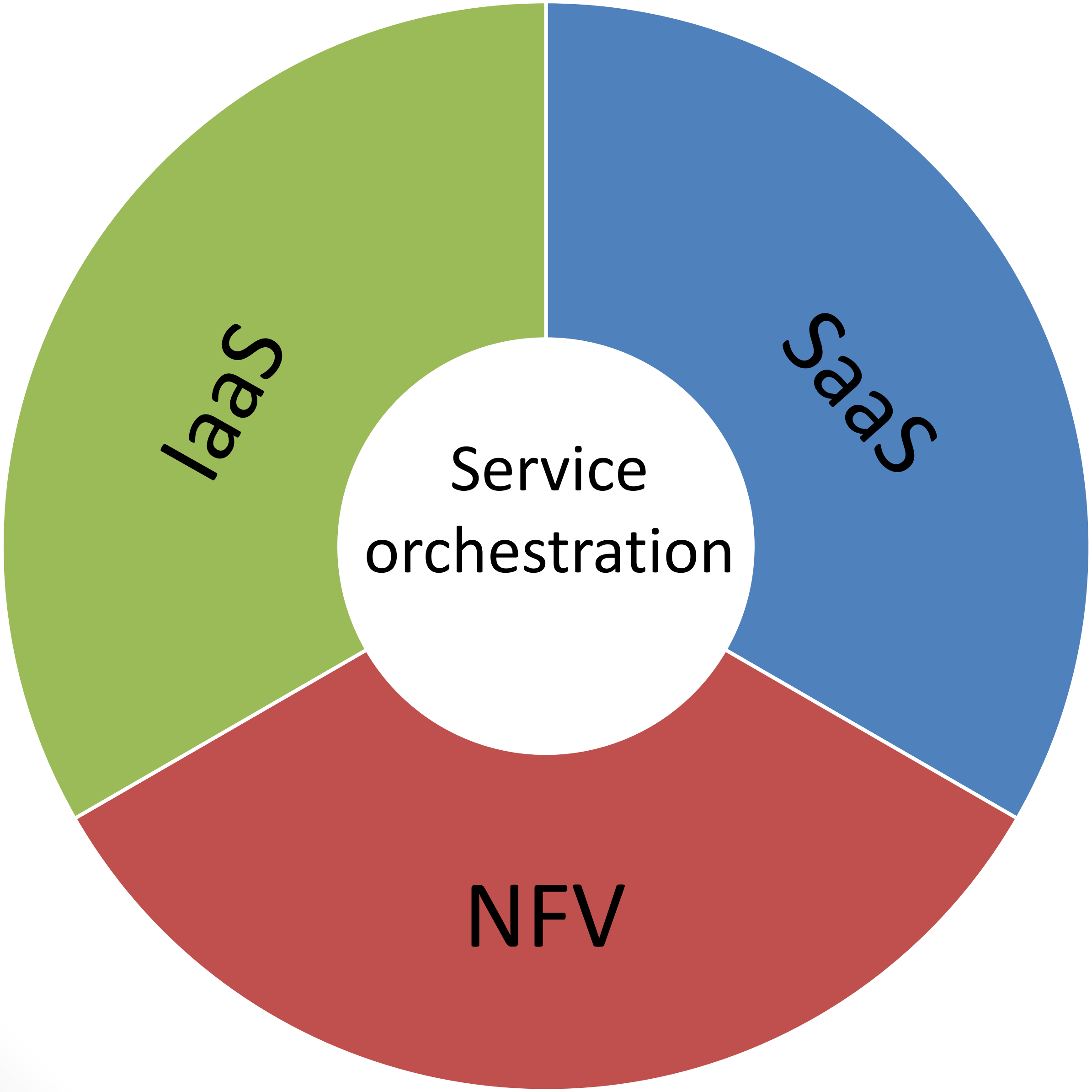
System functional architecture

- Processing of genomic data done in virtual machines (VMs) in cloud data-centers
  - Genomic pipelines VM
- File distribution assisted by caches
  - Distributed NFV paradigm
    - Deployed in routers and VM in data centers
- Processing request managed via easy-to-use web interface
- Central manager for service orchestration (genomic computing manager, GCM)

The big picture (1/2)

- Overall service framework is composed of:
  - An IaaS infrastructure
    - to handle the computing in data centers
  - A SaaS model
    - to interact with the end users (biologists/doctors)
  - An NFV-based content caching distribution
    - to mitigate the impact of huge content transfers on the network

The big picture (2/2)



Involved technologies (1/2)

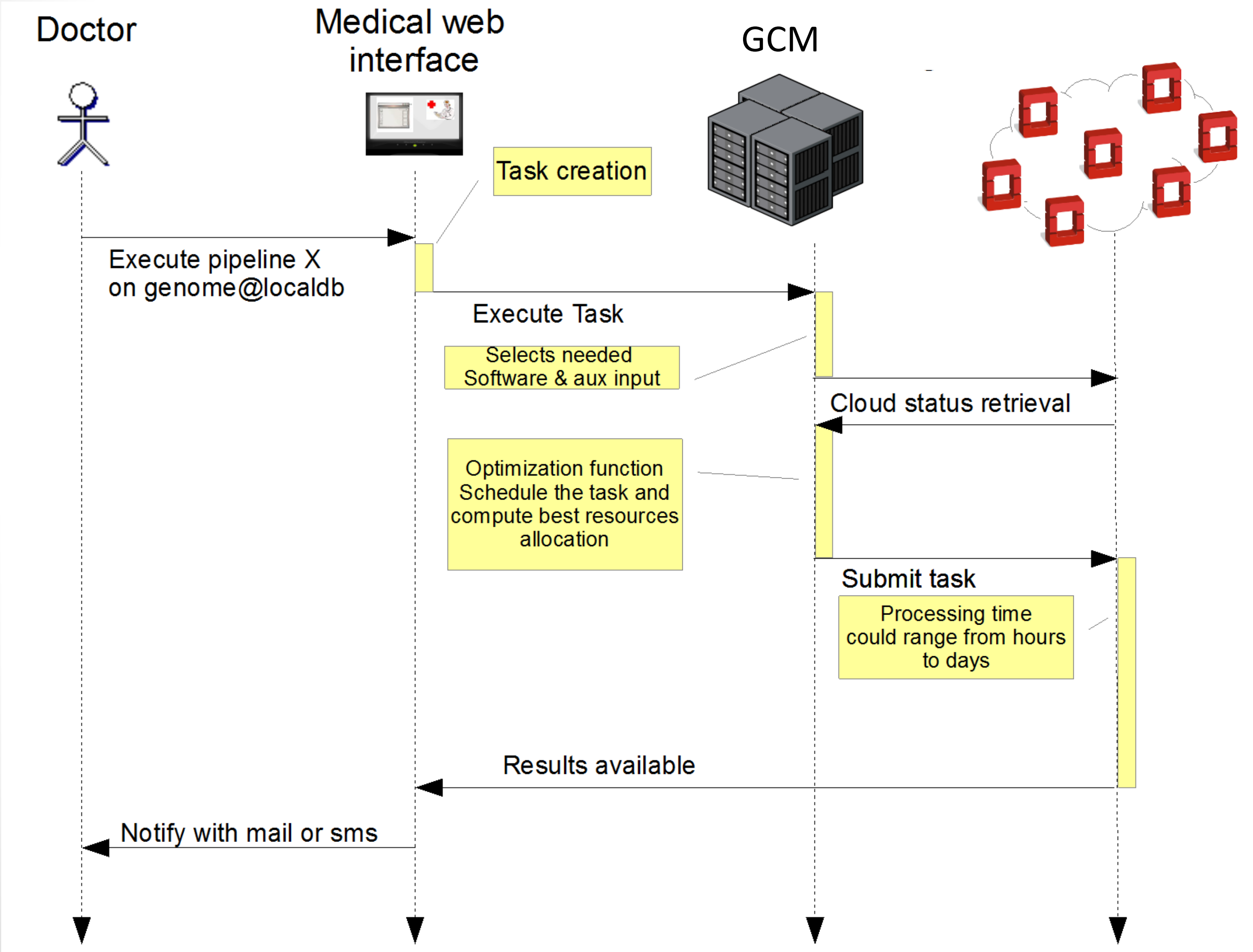
- IaaS:
  - OpenStack as cloud management system
  - KVM as hypervisor
- SaaS:
  - Web interface to interact with users
- NFV
  - Caching function
  - Service virtualization platform: NetServ*
    - Deployable in VM, also working with SDN

*M Femminella, R Francescangeli, G Reali, JW Lee, H Schulzrinne, An Enabling Platform for Autonomic Management of the Future Internet, IEEE Network Magazine 25 (6), 24-32

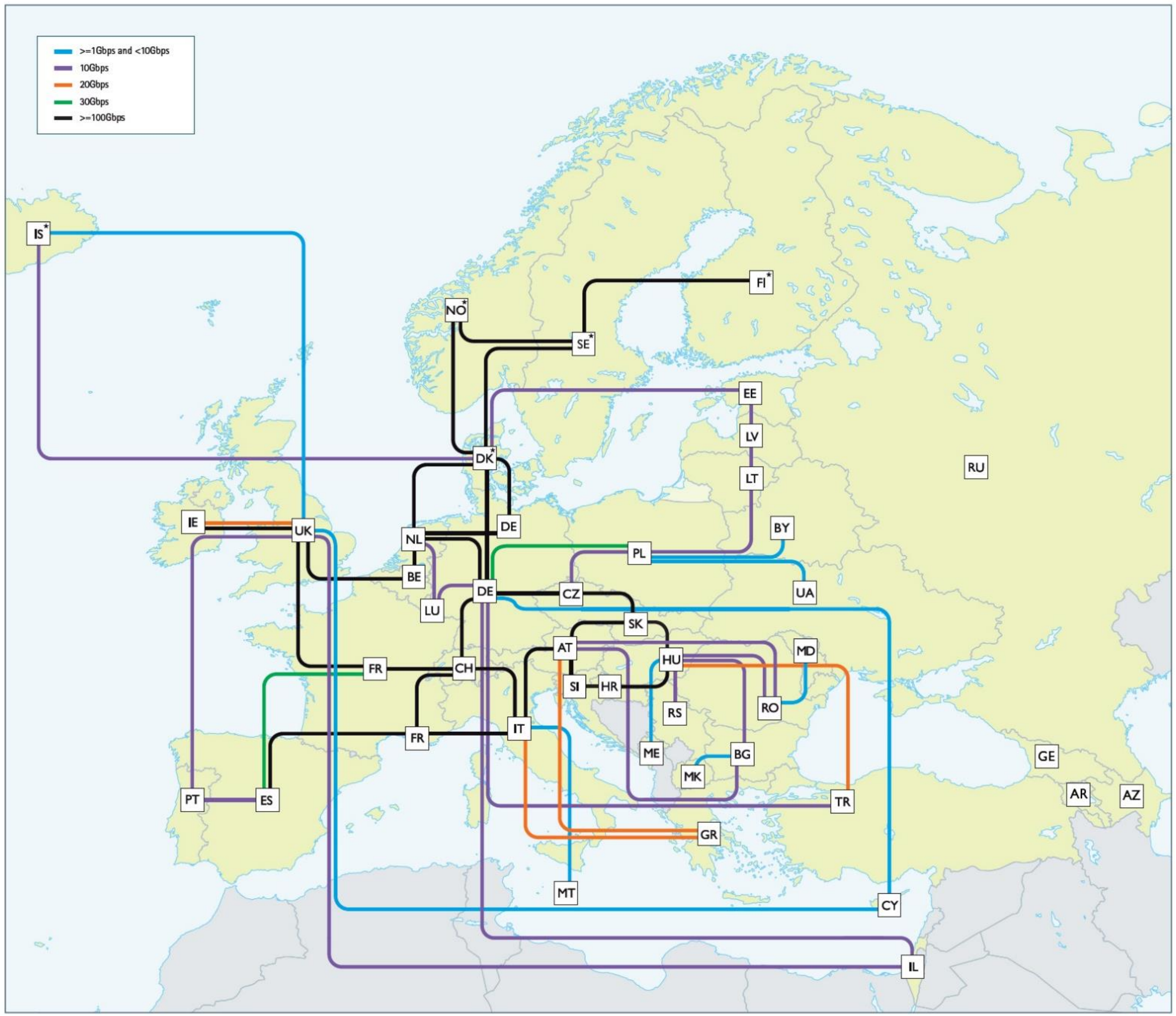
Involved technologies (2/2)

- An orchestration engine (GCM) in Java
  - Able to drive tenants' operations interacting with locally deployed plug-in in data centers
- A novel signaling solution: off-path signaling protocol (OSP)
  - able to find resources on the overlay

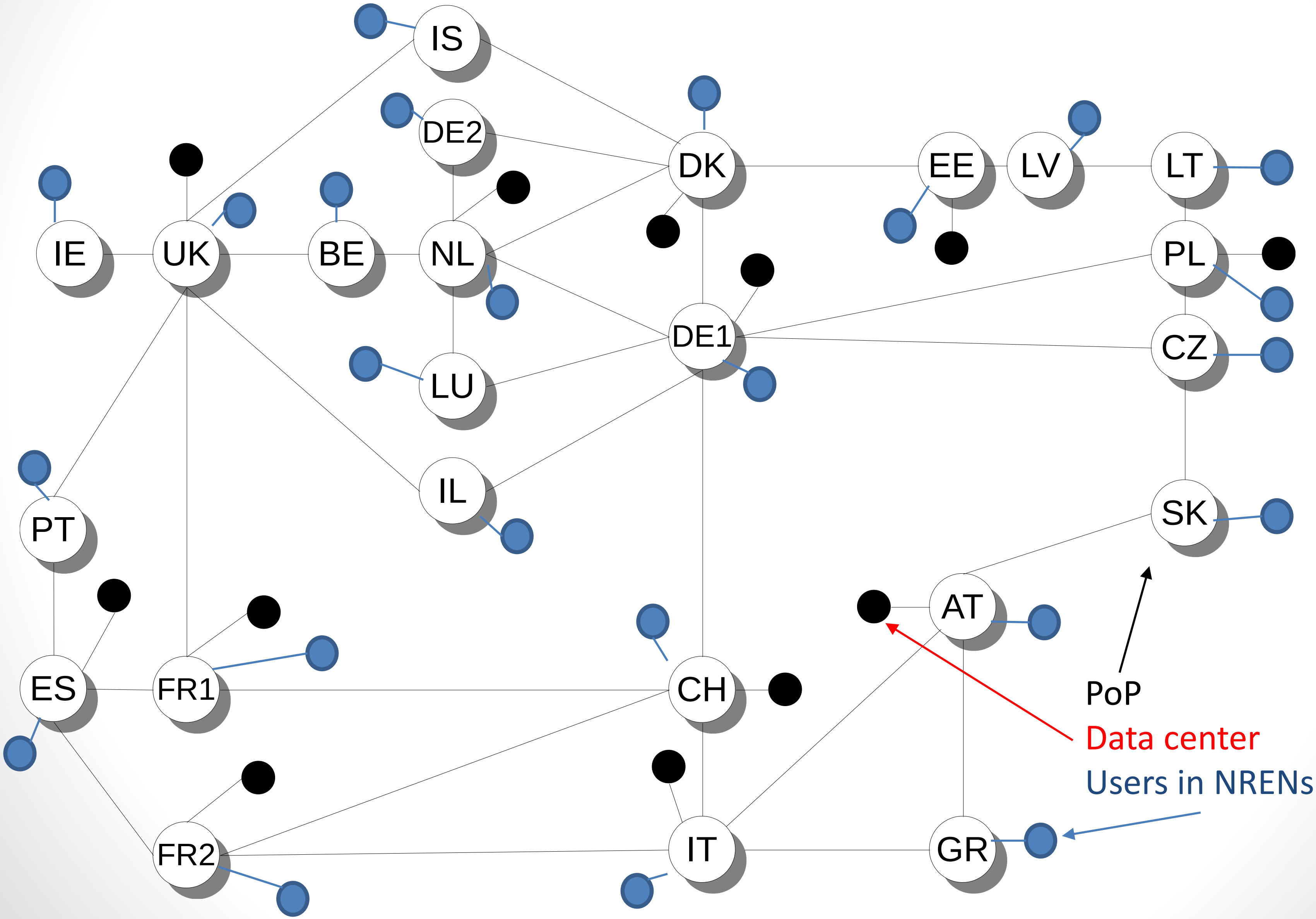
High level signaling



Physical Géant topology



Logical overlay topology



What to test with GTS? (1/3)

- Our dream
  - Implementing in a number of Géant PoPs (in GTS) a full fledged version of ARES
    - A VM for each router
      - At least 2 CPU cores (4 is better) not overbooked, 8 GB RAM, 120 GB of storage ☺
    - An OpenStack tenant for each datacenter
      - Access to physical servers ☺
      - A new testing model (tenant-as-a-service) ☺
      - A set of VMs with many CPU cores (>8), GB of RAM (at least 64 GB), and storage (at least 400 GB) for running a virtualized OpenStack deployment ☺
        - **virtualization inside virtualization is the evil**
    - Real users requesting processing of genomic data sets
  - Not possible, too many resources! ☹

What to test with GTS? (2/3)

- Our desire
  - Implementing the whole network in GTS, emulating the genomic processing but testing real data transfer
    - Running a number of tests to characterize CPU, RAM, and storage requirements of a pipeline VM in our lab
      - Done!
    - Topology consisting of
      - Routers with cache modules
        - At least 2 CPU cores (not overbooked, better 4), 8GB of RAM, 120 GB of storage 😲
      - VMs with cache and management modules
        - At least 2 CPU cores (not overbooked, better 4), 8GB of RAM, 120 GB of storage 😲
      - VM to emulate users sending processing requests
        - Standard GTS VM is enough ☺
  - Still not possible, too many resources! ☹

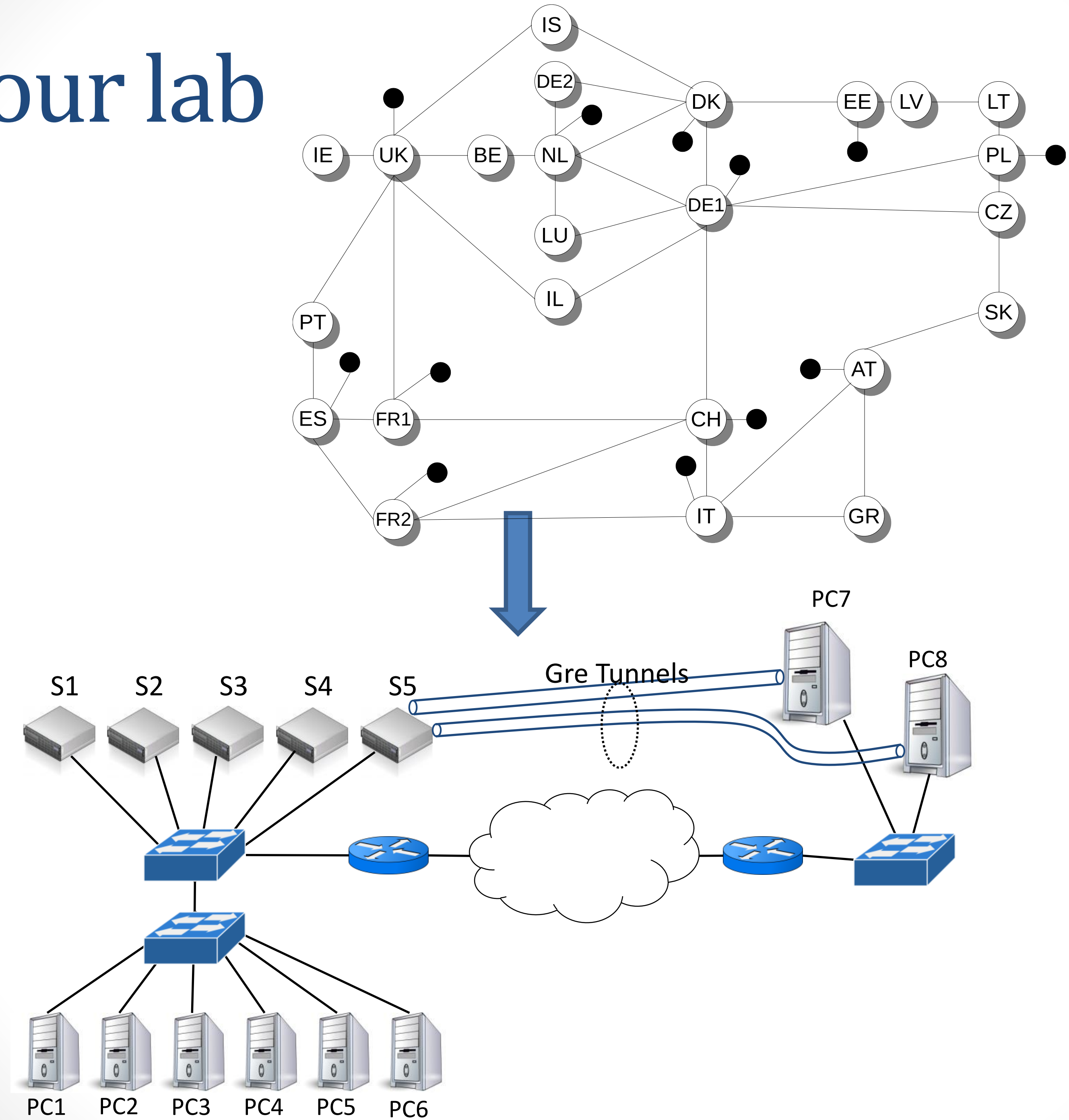
What to test with GTS? (3/3)

- Backup
  - Implementing the testbed topology
    - Routers: 1 standard GTS VM
    - Datacenter: 1 standard GTS VM
    - No users
  - … and testing just the OSP signaling
    - Better than nothing
    - But topology generation is almost a nightmare ☹
    - The version of VM is slightly different than the one used in our testbed 😕
      - It has an impact on the OSP protocol implementation
  - Not done, we have a smarter idea 😃

What we have done

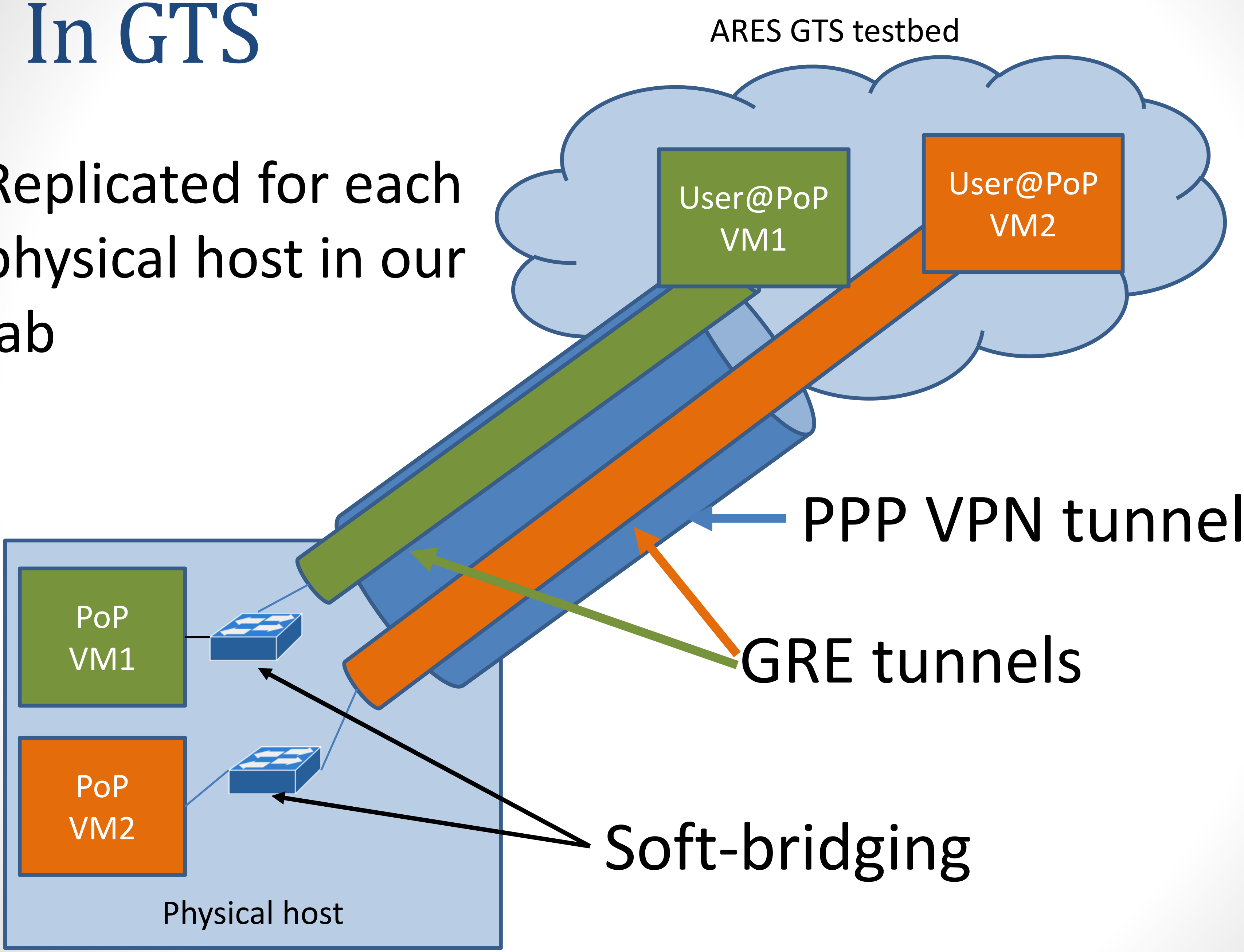
- We have realized a hybrid testbed 😎
  - Core of the overlay is in our lab
    - In our lab
      - 8 PCs and 5 servers well provisioned
        - VMs emulating routers with caches
        - VMs emulating OpenStack tenants + caches
    - Not enough CPU resources on GTS VMs
  - Users requesting processing in GTS
    - A VM for each user
      - Off-load part of the computing burden from our testbed (~30% of nodes)
      - Test data transfers with realistic latency
      - Users submit many GBs of input files to the network

In our lab

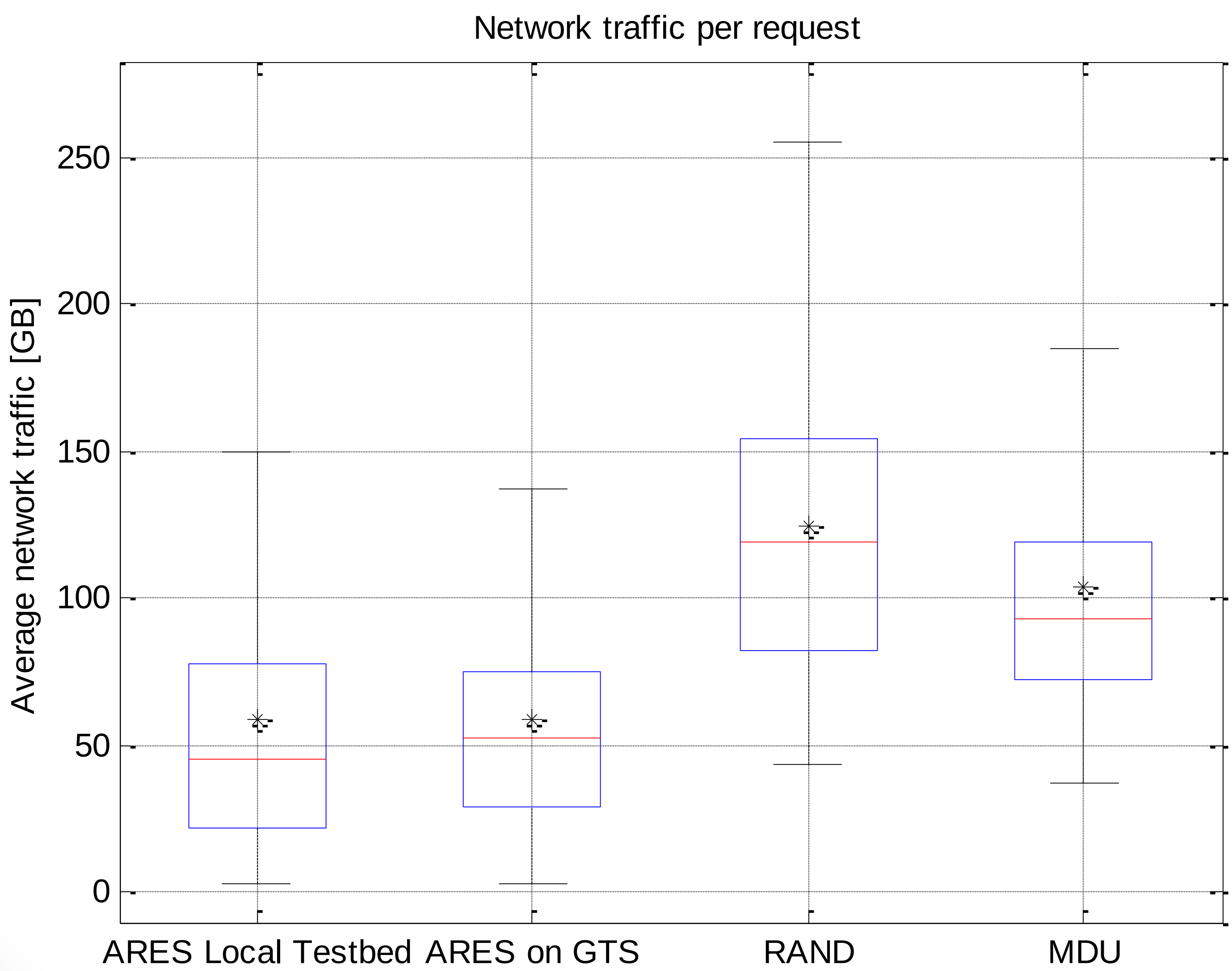

IS
DE2
DK
EE
LV
LT
IE
UK
BE
NL
PL
DE1
CZ
LU
IL
PT
SK
AT
ES
FR1
CH
FR2
IT
GR
PC7
PC8
Gre Tunnels
S1
S2
S3
S4
S5
PC1
PC2
PC3
PC4
PC5
PC6

In GTS

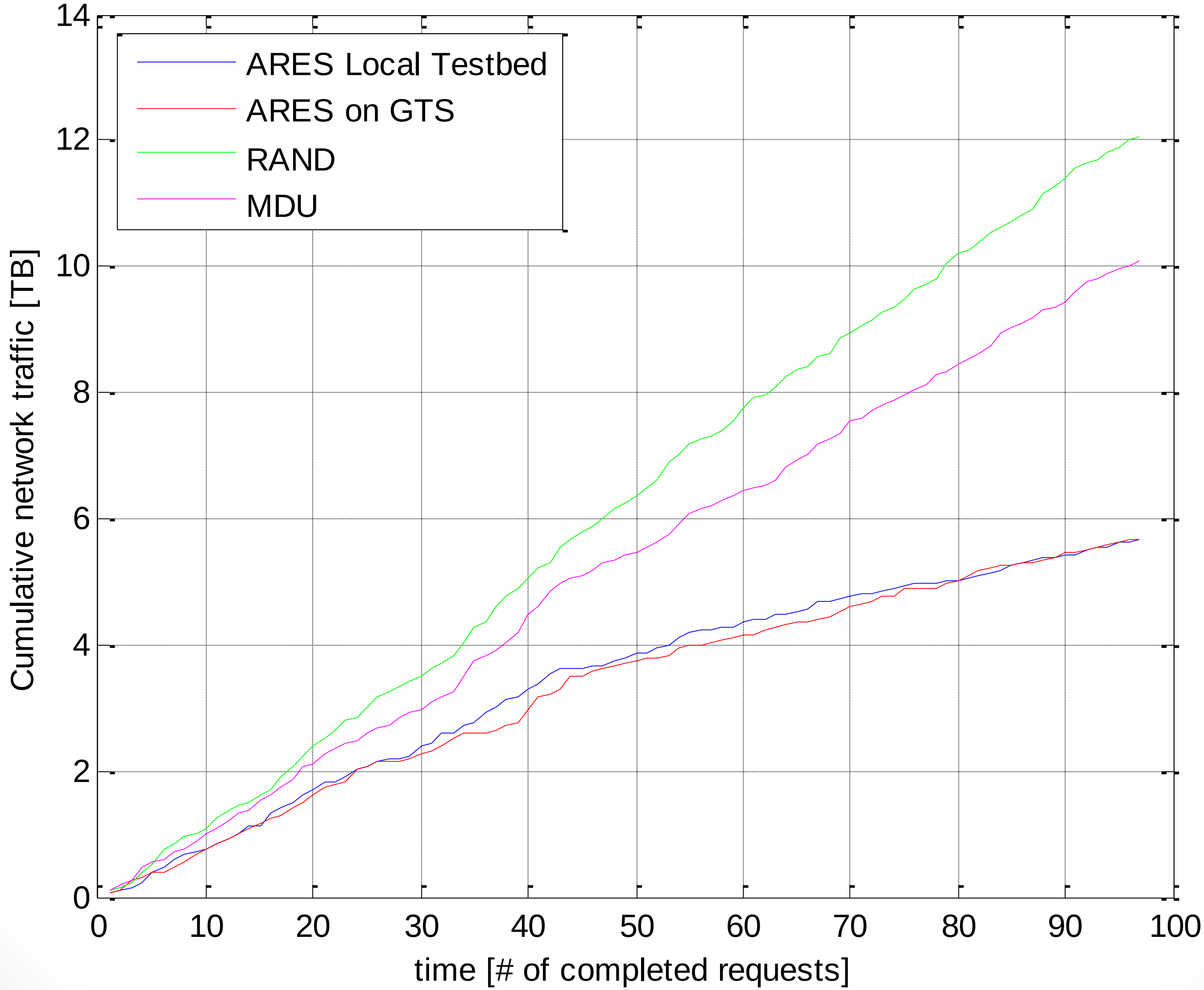
Replicated for each physical host in our lab



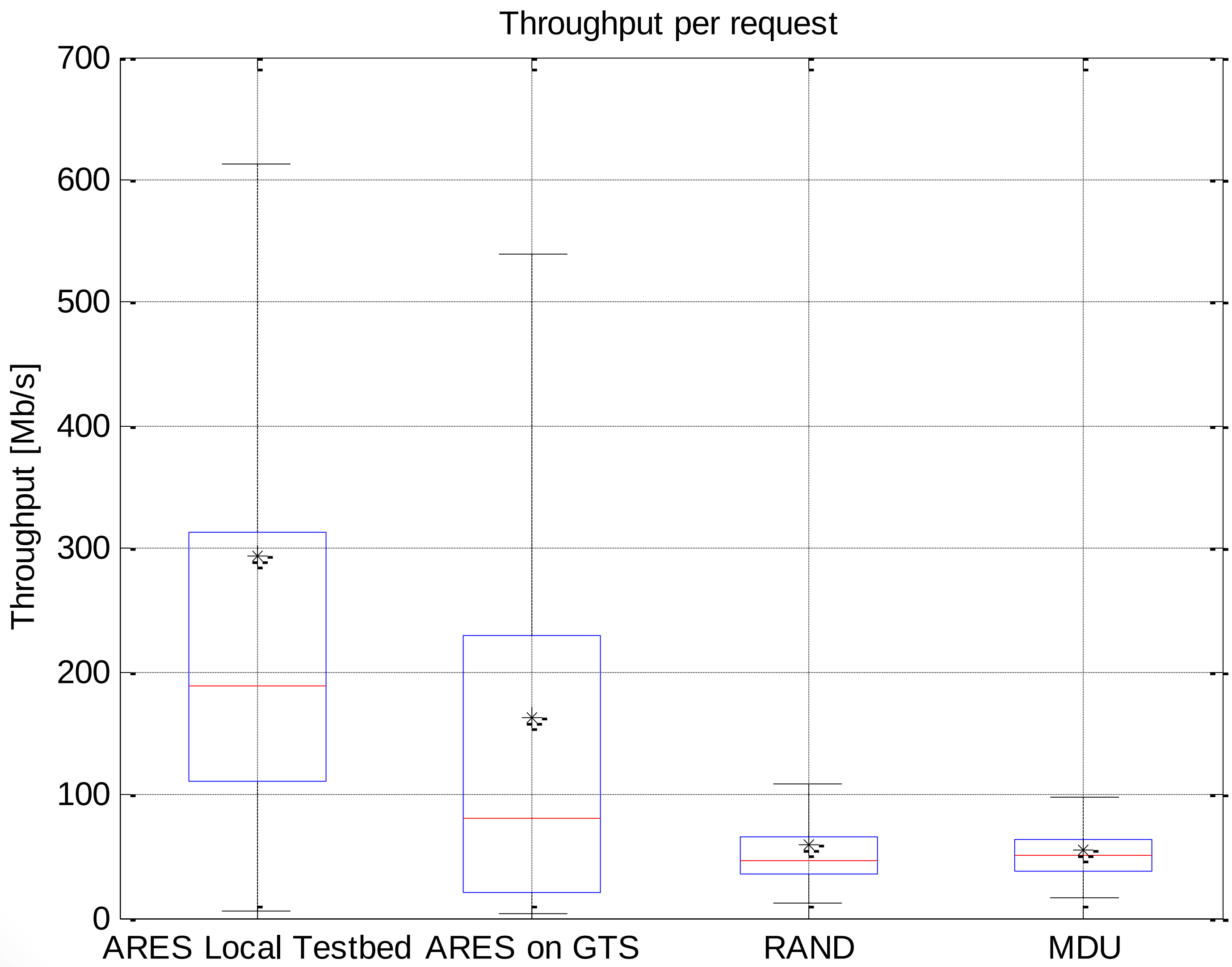
Performance evaluation (1/3)



Performance evaluation (2/3)



Performance evaluation (3/3)



Conclusion

- Our hybrid lab-GTS deployment allows to test a number of interesting features
  - Adding real WAN latency to data transfers
  - Compete for bandwidth with usual traffic
    - As it should be in a hypothetical deployment in Géant datacenters
    - UniPG has 1 Gb/s connection towards GARR NREN
  - Off-load our testbed from many VMs
    - No testbed crashes during hybrid functioning
    - A lot of crashes during full local deployment
  - Validate results obtained in our lab

Open issues (1/2)

- Deploying a complex network is really time-consuming
  - Naming an interface «ethX» in DSL does not map into «internal ethX» interface in the VM
  - A tool for designing complex topologies is definitely needed!
- VM size cannot be configured
  - Number of CPU cores
  - Amount of RAM
  - Disk size
    - Choice is present but not active up to some time ago

Open issues (2/2)

- Possibility of using a custom VM image would help a lot
  - Saving of configuration and installation time
  - Avoid compatibility issues of developed software
- IAG VPN not so stable
  - We experienced some crashes of the PPP VPNs, not of the VM
  - We have not tested the External Domain facility
- A front-end scheduler is required for massive usage
  - How many resources are available to each user at a given time
  - Possibility of scheduling VMs of variable size