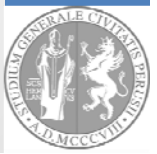
A Resource Discovery Framework for Cloud-based Genomics Computing

Mauro Femminella, G. Reali, D. Valocchi, E. Nunzi
University of Perugia
mauro.Femminella@unipg.it

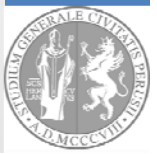
Summary

- Genomic processing scenario
- System architecture
- The signaling framework
- Performance evaluation
- Conclusion and future work



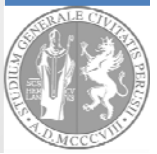
Genomic computing scenario

- Genomic data sets is a special case of scientific computing (Big^2 data):
  - the number of available genome files is becoming extremely large
    - drop in the genome sequencing costs
  - each individual data set is significantly large, in the order of tens of GB
- A wide diffusion of cloud-based genomic data processing will have a significant impact on network resources
  - each processing request will require the transfer of tens of GBs into computing nodes



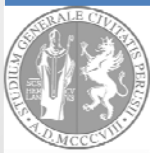
System functional architecture

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

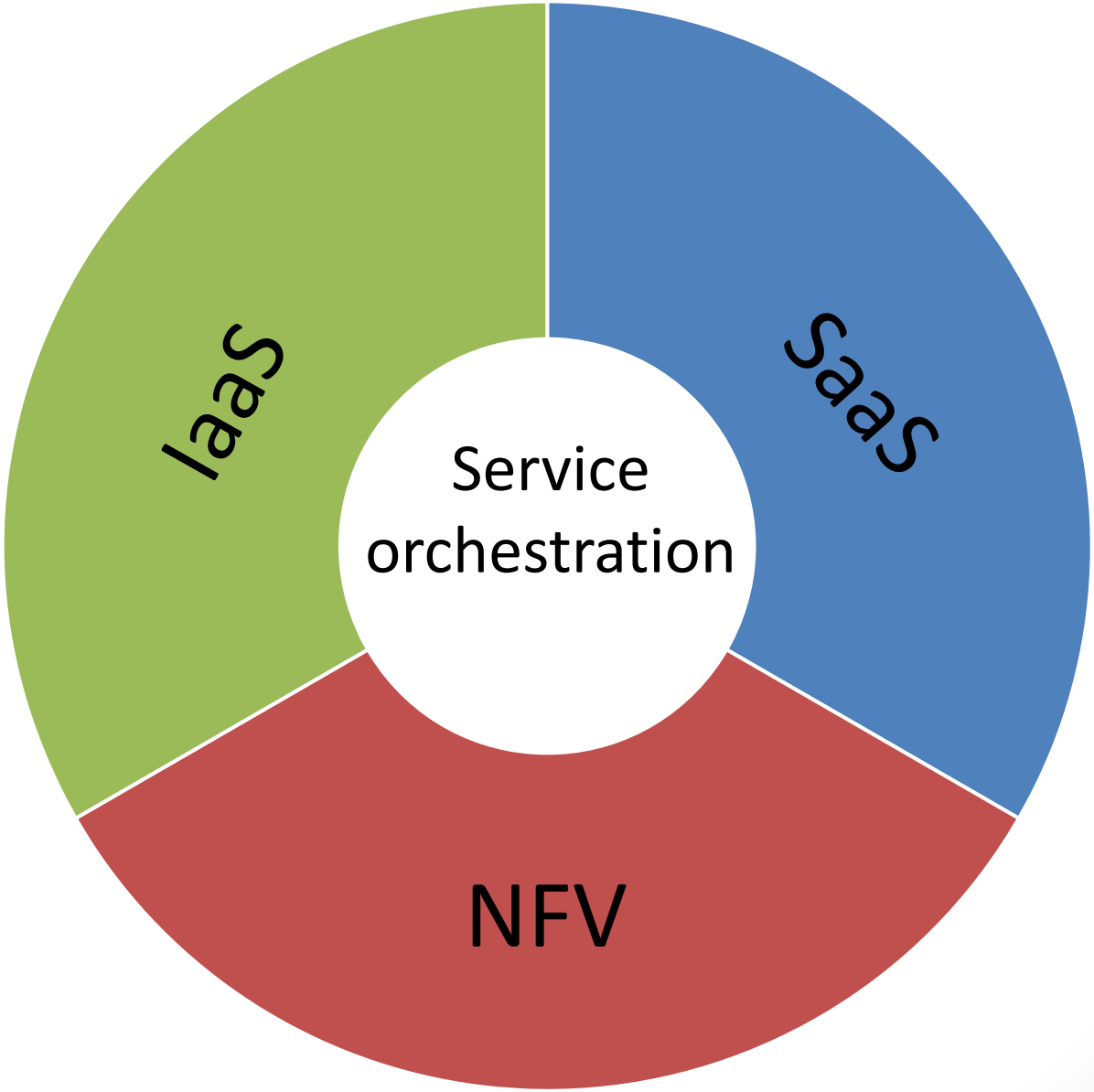


The big picture (1/2)

- 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



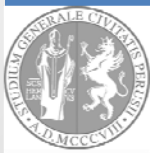
The big picture (2/2)



Involved technologies

- IaaS:
  - OpenStack as cloud management platform
  - KVM as hypervisor
- SaaS:
  - Web interface to interact with users
- NFV
  - Caching function
  - Virtualization platform NetServ*
    - Deployable in VM and routers (Juniper, ongoing)

*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

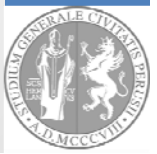


Signaling (1/2)

- But, for orchestration, we need signaling!!
  - Requirement 1: a discovery framework able to provide information on the resource availability
    - Contents in caches
    - Computing resources (storage, memory, CPU) in data centers
  - Requirement 2: we need to identify resources as much as close to the IP path connecting involved parties, since we are moving GBs of data each computing session
    - We propose to perform the search within a hose including the IP path connecting involved entities (off-path signaling)
      - VM repository
      - Auxiliary genomics files repository
      - Private database hosting the genome to process, or public database hosting the anonymized genomics to process

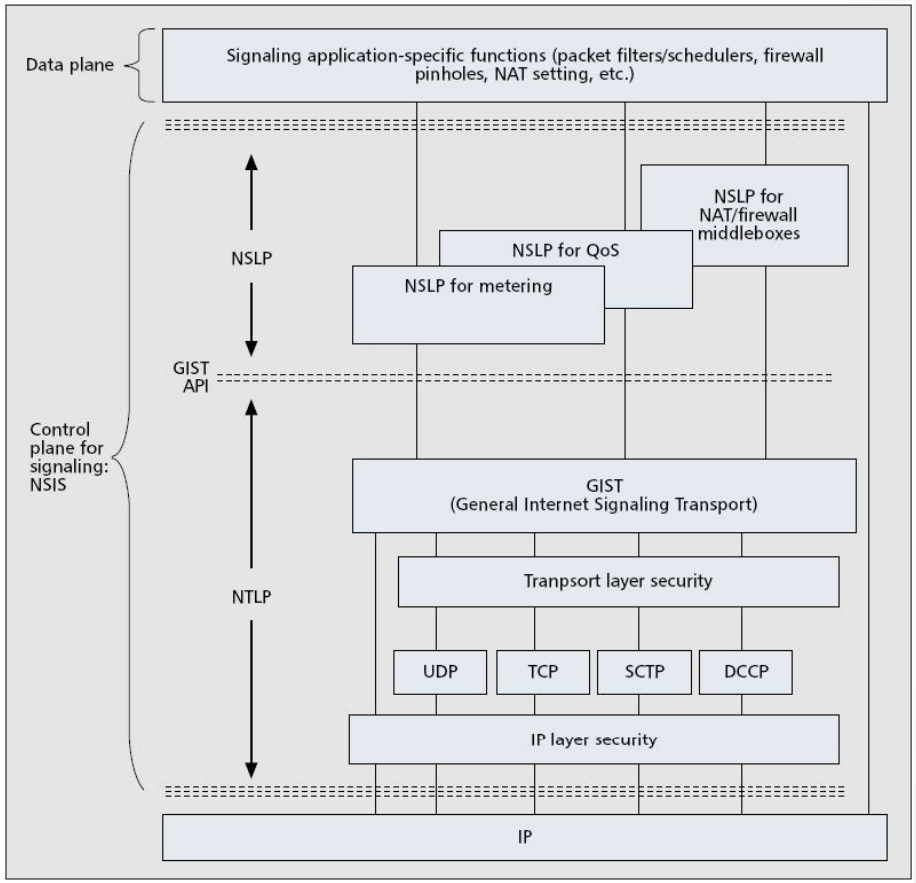
Signaling (2/2)

- Our choices:
  - Web based signaling (RESTful model)
    - For interactions with users
    - For point-to-point communications
  - IETF NSIS signaling
    - Why NSIS? Because it has on-path interception as native function
    - For discovering resources in the neighborhood of a path (hose model)
    - The radius of the hose is a signaling parameter
    - NSIS is used as signaling platform for NetServ
      - We designed a specific NetServ bundle for handling the communication with OpenStack
        - No need to install NSIS inside OpenStack
        - Plug & play model



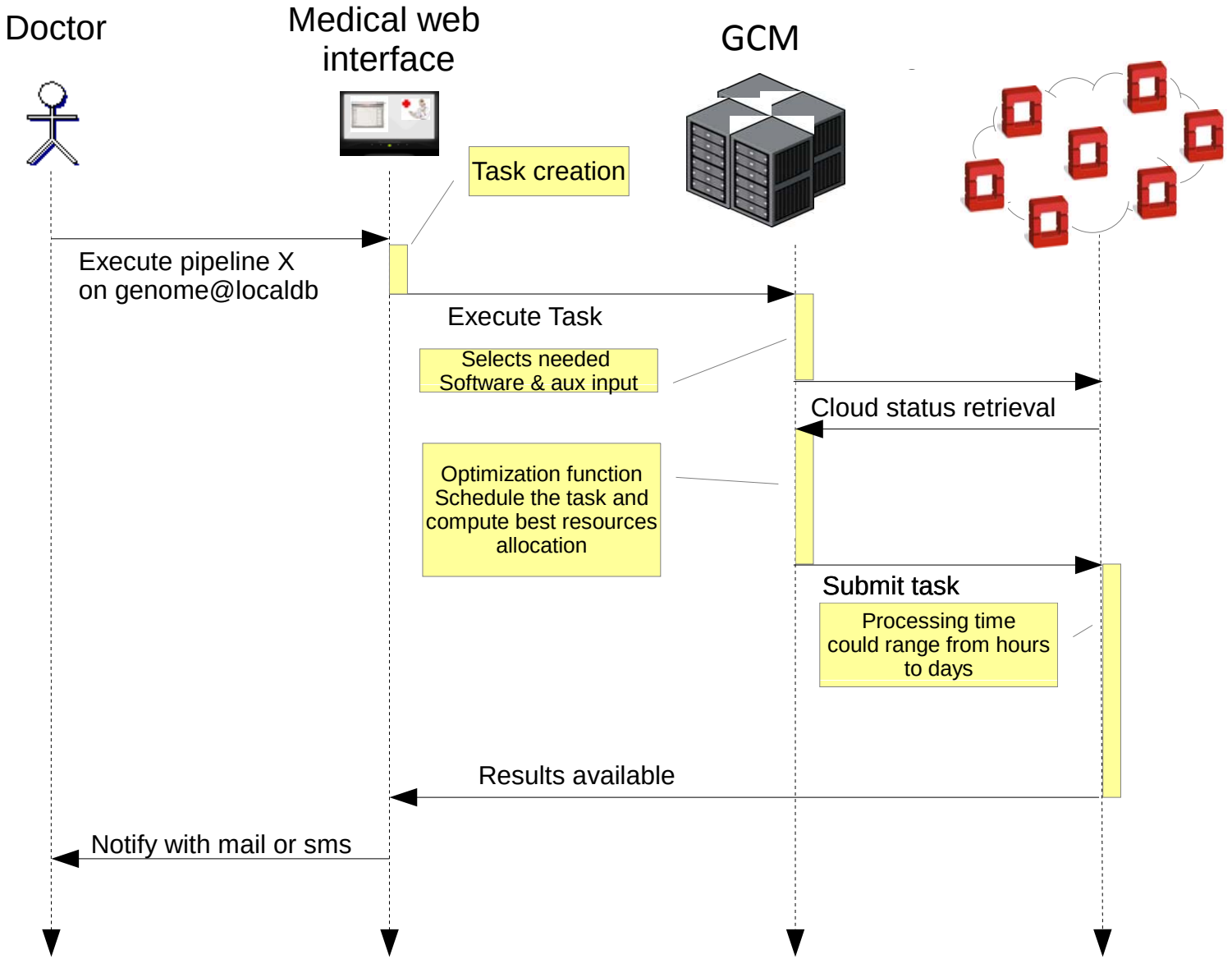
NSIS (Next Steps In Signaling)

- IETF RFC 4080
- Suite of protocols able to support various signaling application
- Two layers:
  - NTLP: NSIS Transport Layer Protocol
    - GIST (Generic Internet Signaling Transport)
    - Extended for handling off-path signaling **
      - Peer discovery and distribution via flooding to neighbors
  - NSLP: NSIS Signaling Layer Protocol
    - Designed on purpose
- Development platform:
  - nsis-ka

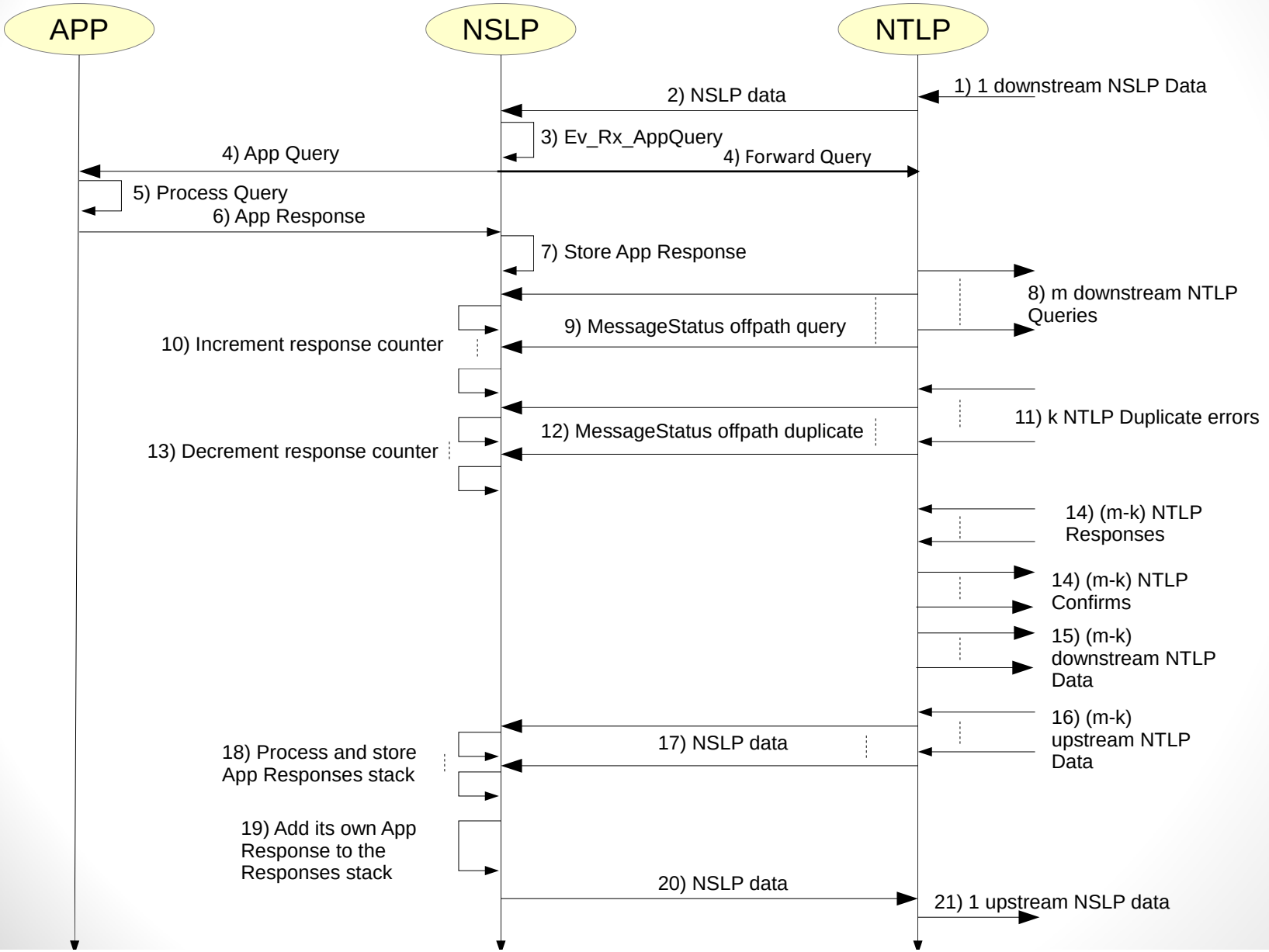


**M Femminella, R Francescangeli, G Reali, H Schulzrinne, Gossip-based Signaling Dissemination Extension for Next Steps In Signaling, IFIP/IEEE NOMS 2012

High level signaling



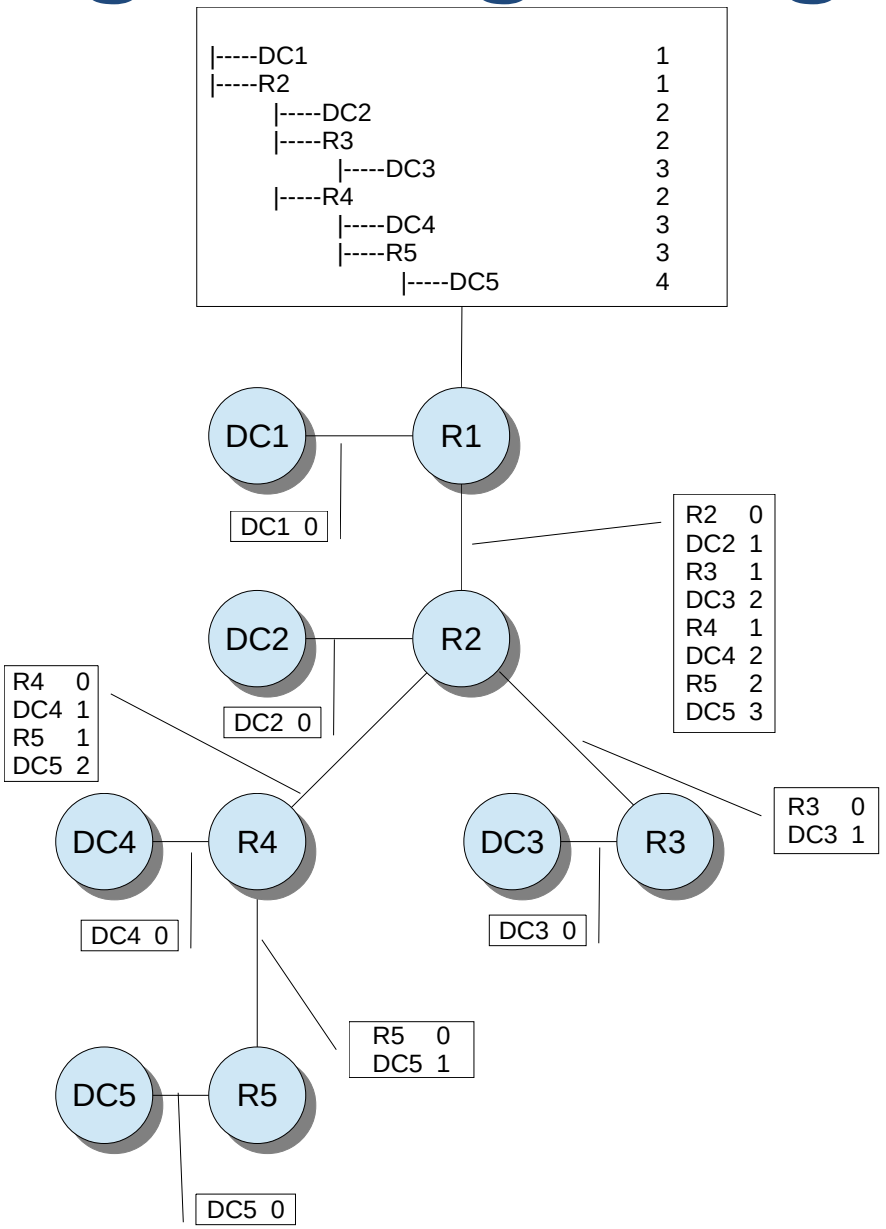
Our signaling



Data structure

Query	=	NSLP-header [Application-requirement list] [Content-Id list]
QueryResponse	=	NSLP-header ResponseWrapper stack
ResponseWrapper	=	common-header Node-Id Depth Response code

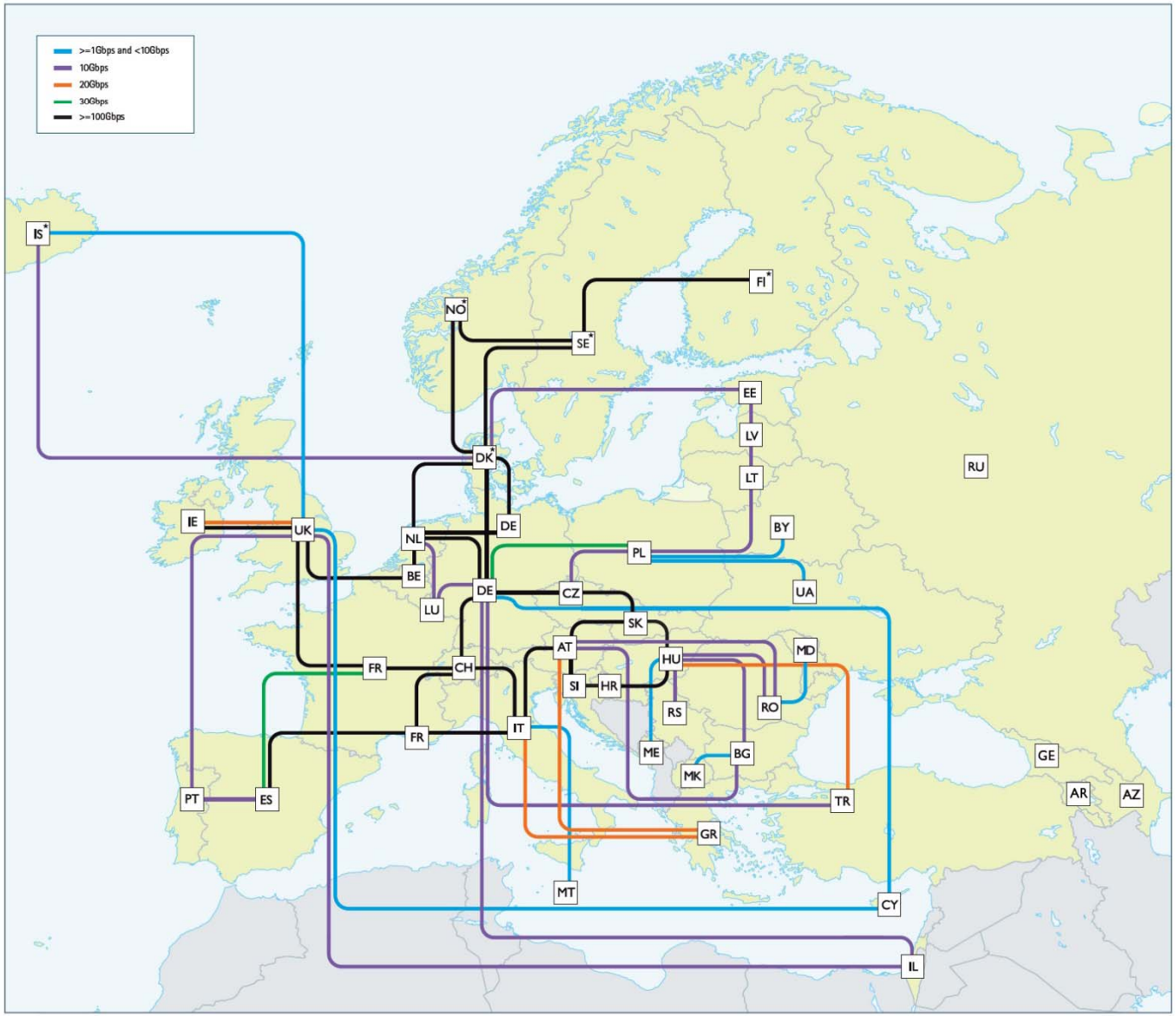
What we get @ signaling source



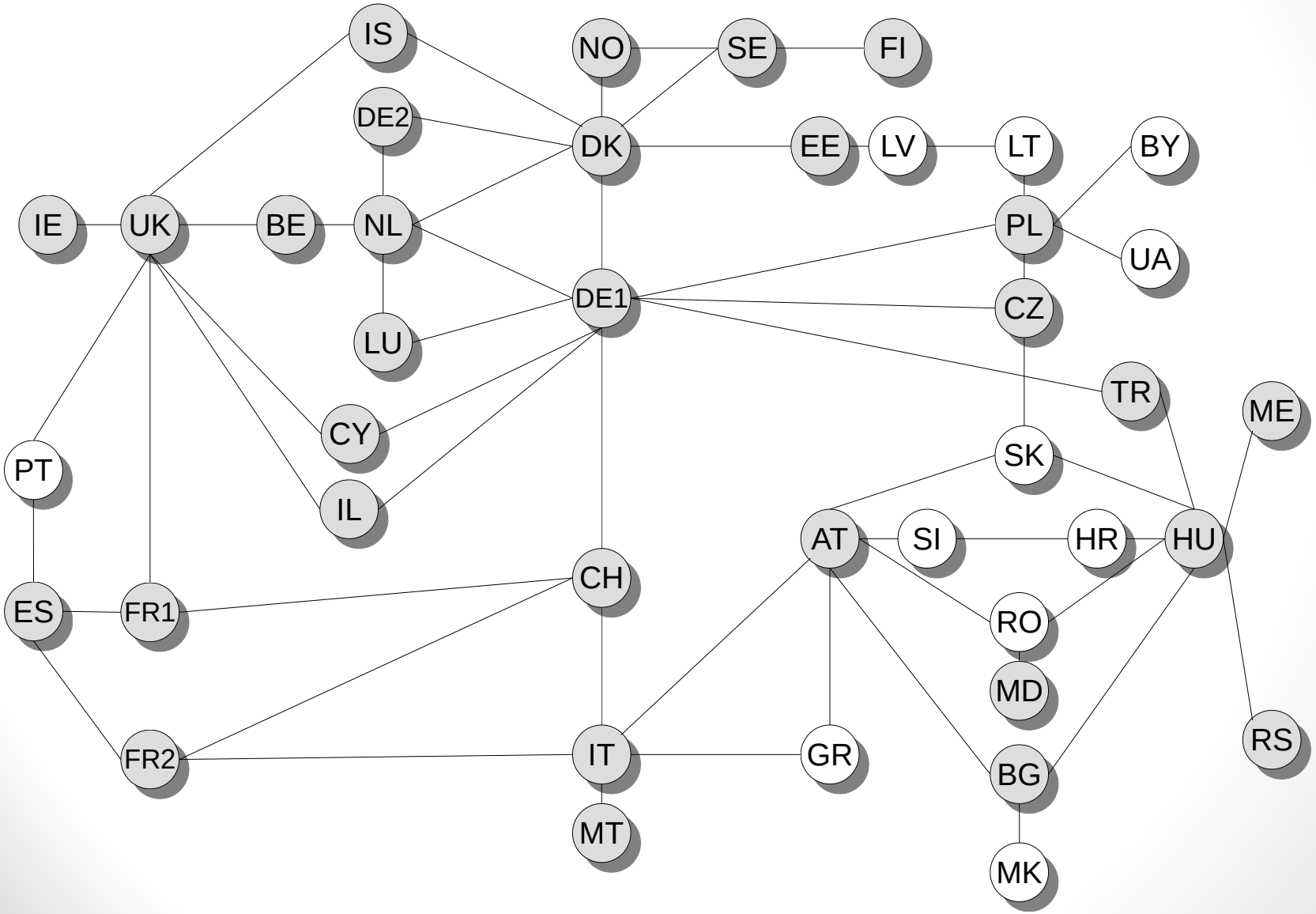
Performance evaluation

- We measure which is the cost of the signaling to the network
  - We have to take into account that we move GBs each session
- We have already proved that an «informed» decision allows to decrease the amount of exchanged traffic by a factor 6
- Tests:
  - We emulated the Géant topology
    - One VM for each Géant PoP
    - One VM for each data center connected with a Géant PoP

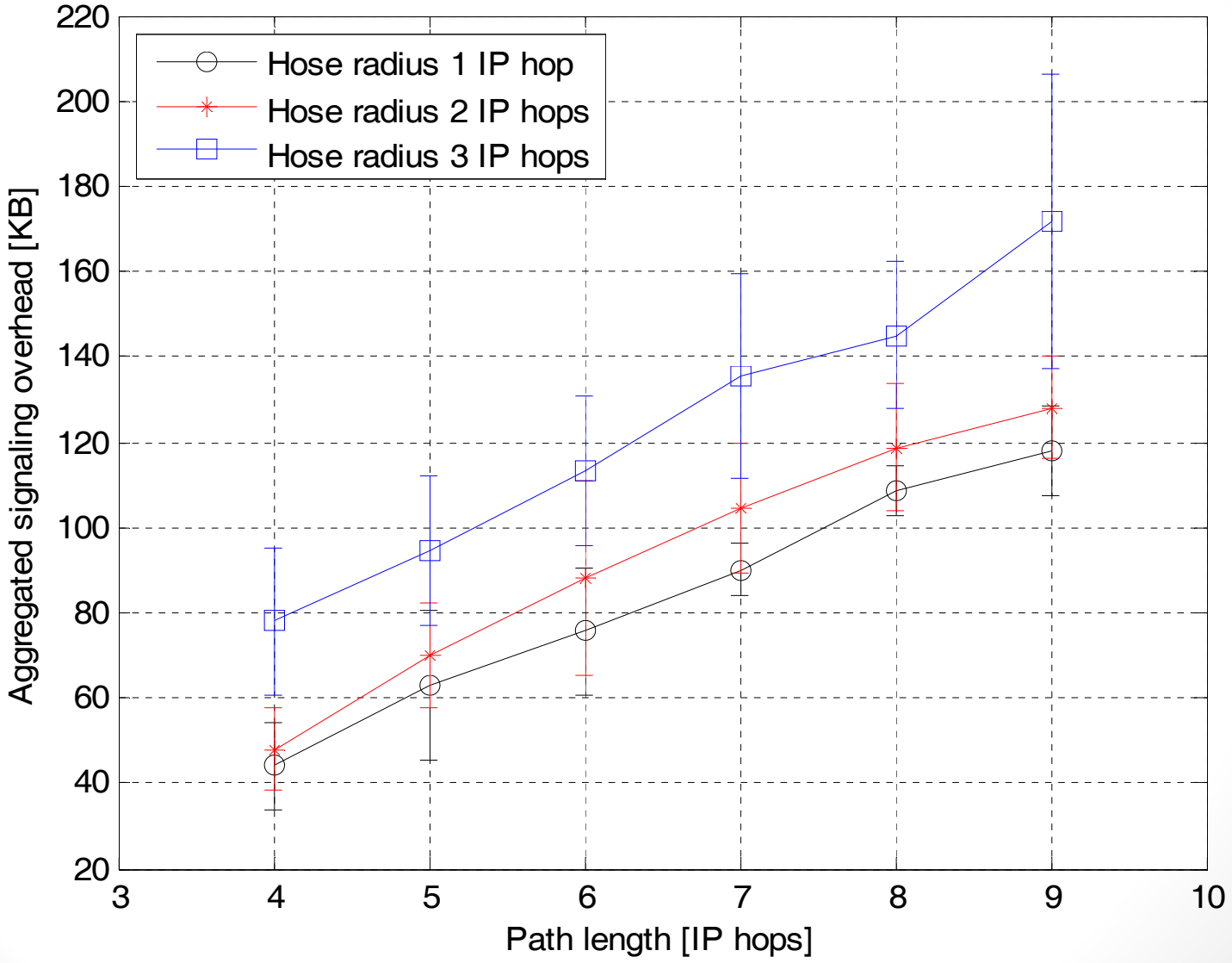
Physical topology



Logical topology



Overhead


Hose radius 1 IP hop
Hose radius 2 IP hops
Hose radius 3 IP hops
Aggregated signaling overhead [KB]
Path length [IP hops]
220
200
180
160
140
120
100
80
60
40
20
3
4
5
6
7
8
9
10

Conclusion

- A signaling framework for resource discovery of cloud resources for genomic processing applications
  - The proposed framework can be used not only to search datacenters able to host the processing, but also caches able to provide the desired content with a lower overhead
  - The specific characteristics of the proposed solution is the capability to provide results with a controlled proximity degree with respect to the data path connecting involved entities
- The results confirm that the signaling overhead is definitely negligible and thus affordable for any network speed
- This solution is applicable also outside the genomic scenarios
  - suitable for all network scenarios in which large amounts of data have to be moved towards cloud sites for later processing
- Future work include a complete system description and deployment