Report on the second year

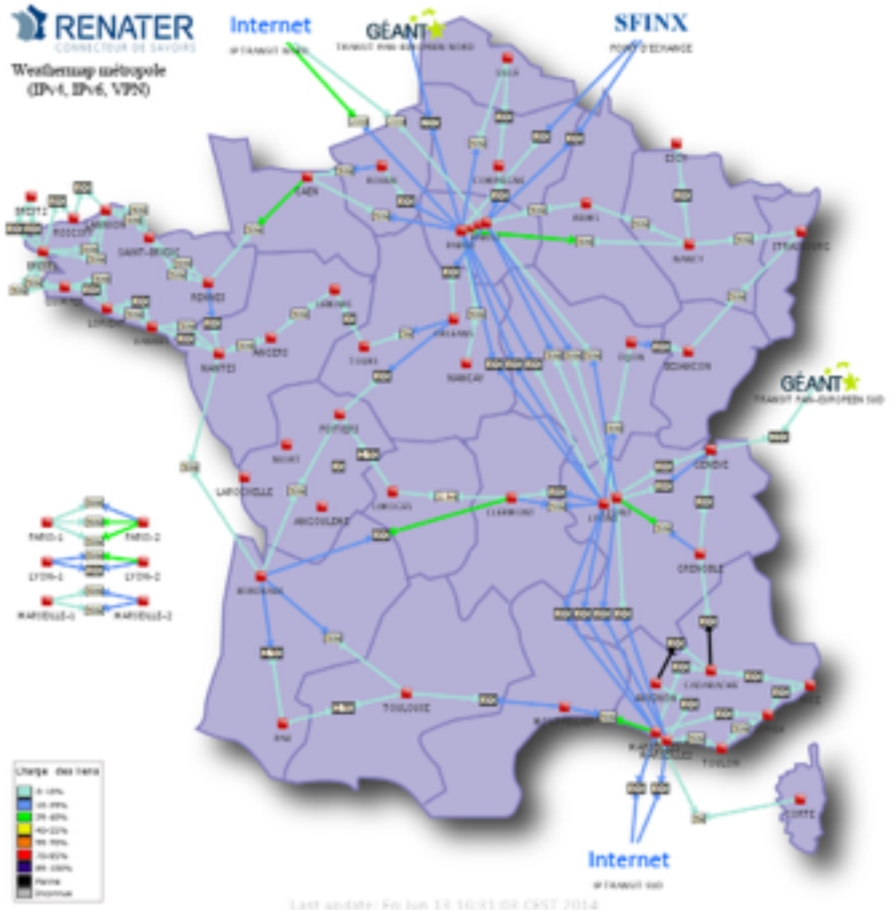
Jonathan Pastor (ASCOLA/LINA/INRIA)
jonathan.pastor@inria.fr

Few facts (1/2)

- Jonathan Pastor.
- Ph.D student under the supervision of F. Desprez and A. Lebre.
- My thesis research started on october 2012.
  (20 months ago)
- Presentation of the work done during the second year of the thesis.

Few facts (2/2)

- Cloud computing has become very popular.
- Ever-increasing demand => ever-increasing infrastructure size.
- PB: scalability, reliability, energy but also security, juridiction and network overhead.
- Decentralise the production of computing ressources (Discovery project, http://beyondtheclouds.github.io/).
  - Leverage the concept of micro/nano DCs [Greenberg2009].
  - Our particularity: deployed and operated on top of Internet backbone [2].
- Design and implement a fully distributed IaaS (LUC-OS).



Agenda

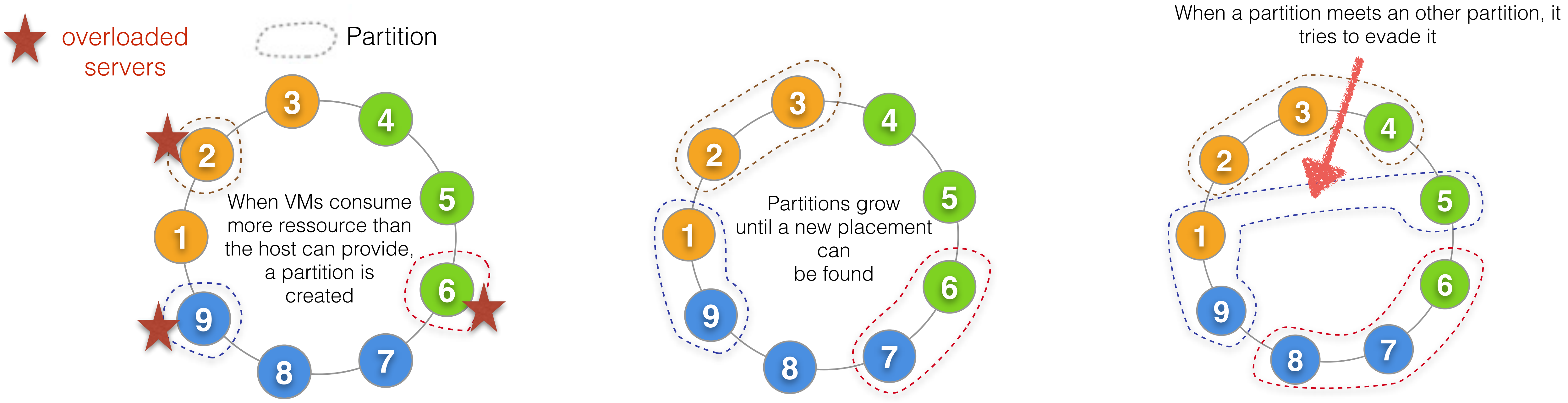
- Back to the first year.
- Contributions during this second year.
- A brief overview of teaching activities
- Ongoing and future work.

First year: validate DVMS

- Distributed Virtual Machine Scheduler.
  (First building block of the LUC-OS, implemented during the Ph.D of Flavien Quesnel)

- VMs located on overloaded servers are migrated on underloaded servers (preserving VMs quality of service).

- Leveraging a **Chord** overlay network.



When VMs consume more ressource than the host can provide, a partition is created

Partitions grow until a new placement can be found

When a partition meets an other partition, it tries to evade it

First year: validate DVMS

- Worked on DVMS:
  - Add overlay networks support to DVMS (PeerActor model).
  - Included fault tolerance in DVMS.
  - Validation of DVMS through simulations on Simgrid.
  - Introduction to Grid’5000 platform and APIs.
- Publication: validation at large scale of the DVMS proposal.
  - [1] Flavien Quesnel, Adrien Lèbre, Jonathan Pastor, Mario Südholt, and Daniel Balouek. **Advanced Validation of the DVMS Approach to Fully Distributed VM Scheduling**. In ISPA' 13: The 11th IEEE International Symposium on Parallel and Distributed Processing with Applications, Melbourne, Australia, July 2013.

Second year work

- Research Report/ Book chapter on the Discovery initiative's objectives [2].
- Introduction of the locality properties concept
  - Integration in DVMS [3].
  - Large scale experiments (grid'5000 challenge) [4].
- Architecting Discovery over OpenStack [5].

Introduction of the locality properties

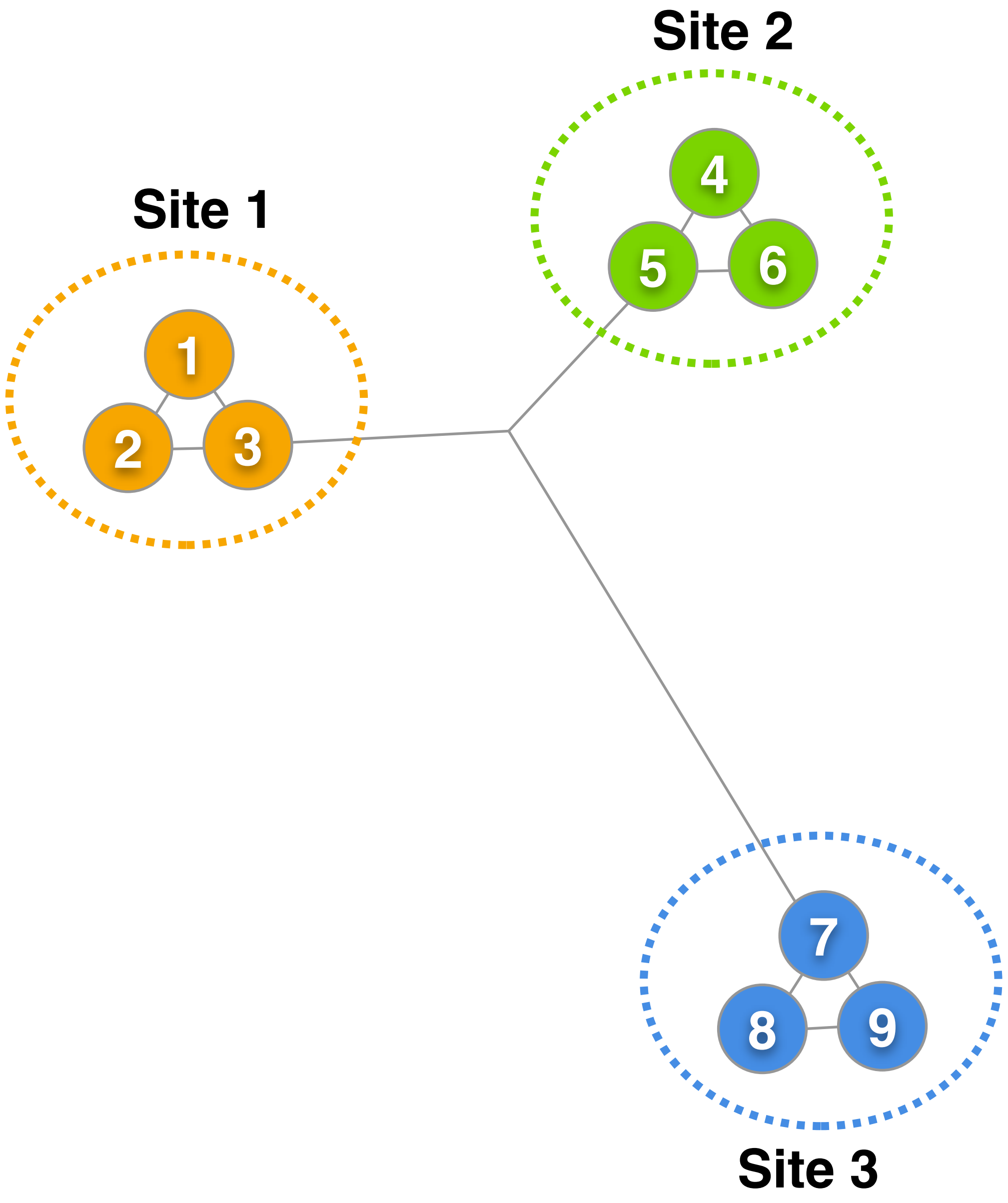
Locality properties

- Discovery: leverage the concept of micro/nano datacenters [Greenberg2009] geographically spread.
  => *nodes can be far from each other.*

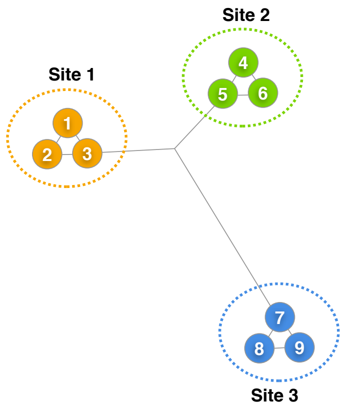
- And we want to maximise cooperation between close nodes/micro DCs.

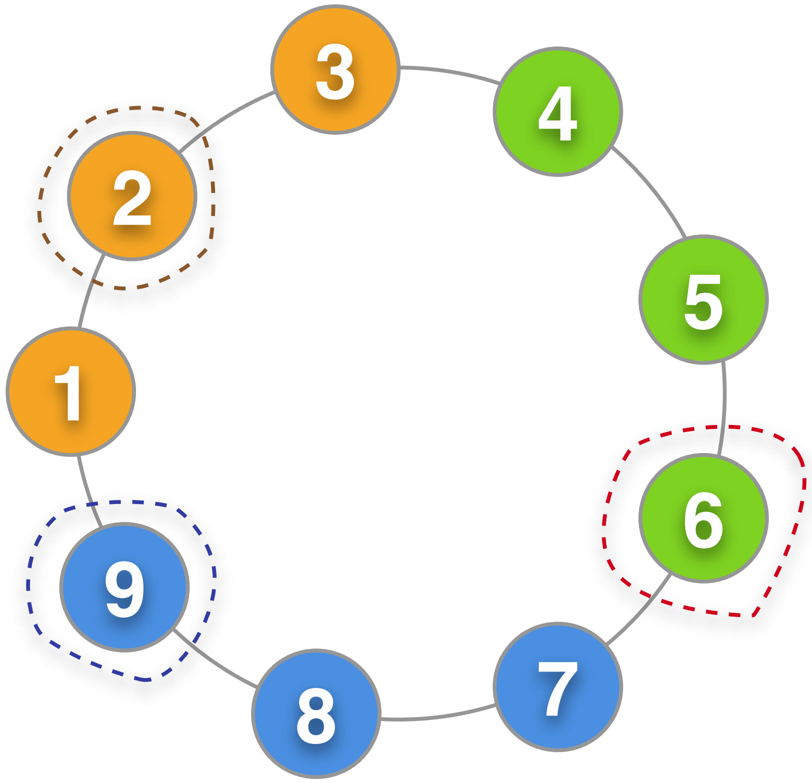
- Example: The DVMS case.
  - The cost of a migration depends of networking parameters (bandwidth and latency).
  - Promote migrations between close nodes.

Example



Chord cannot promote collaboration between close nodes

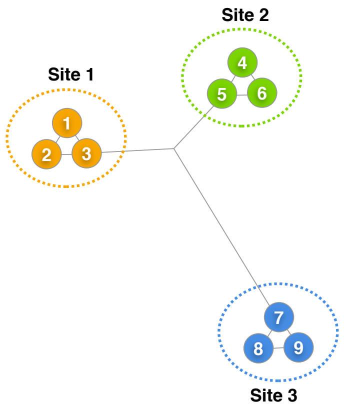


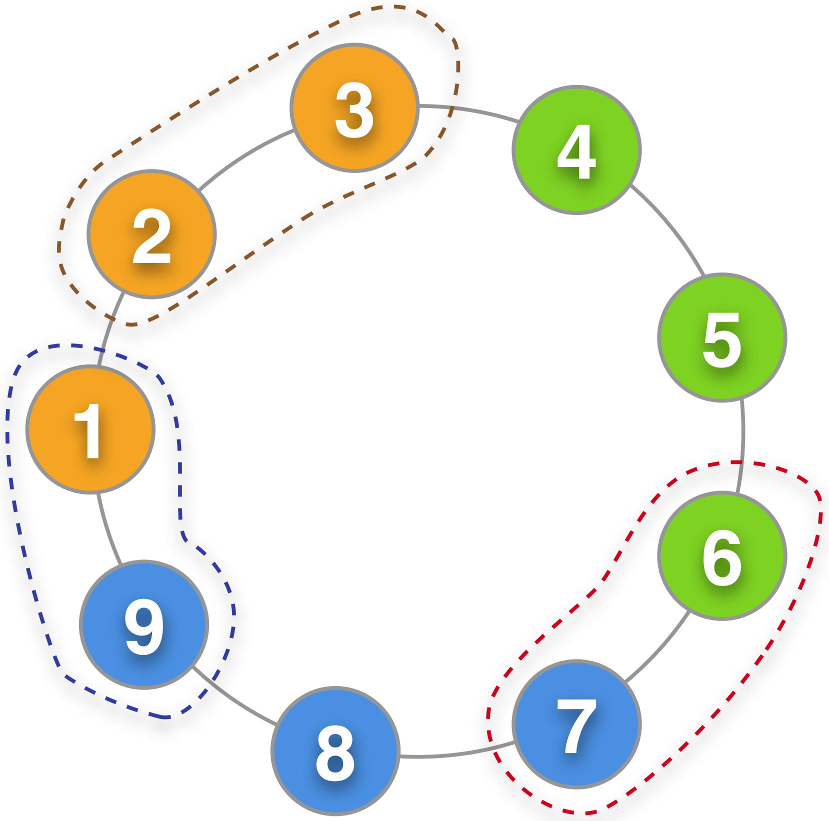


⬭ Partition

⚠ Inefficient collaboration

Chord cannot promote collaboration between close nodes

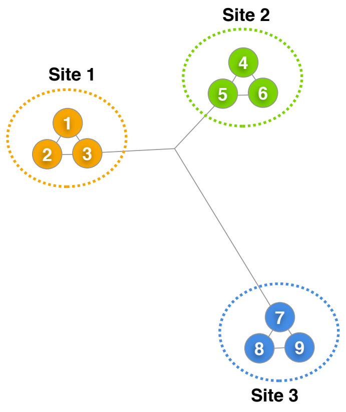


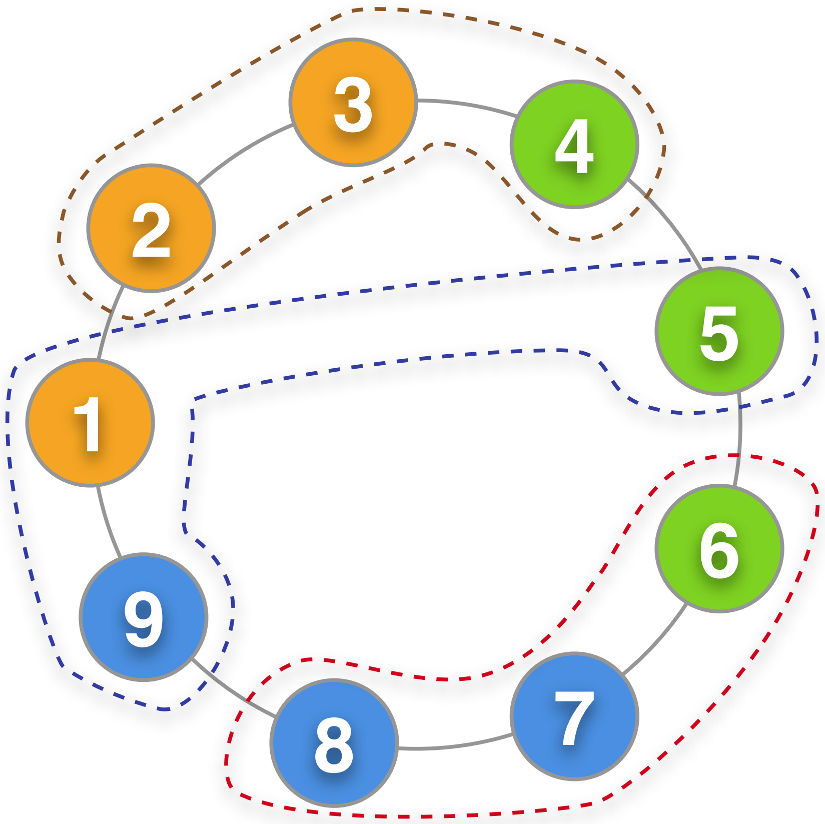


⬭ Partition

⚠ Inefficient collaboration

Chord cannot promote collaboration between close nodes

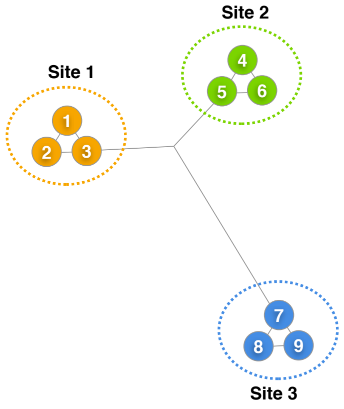


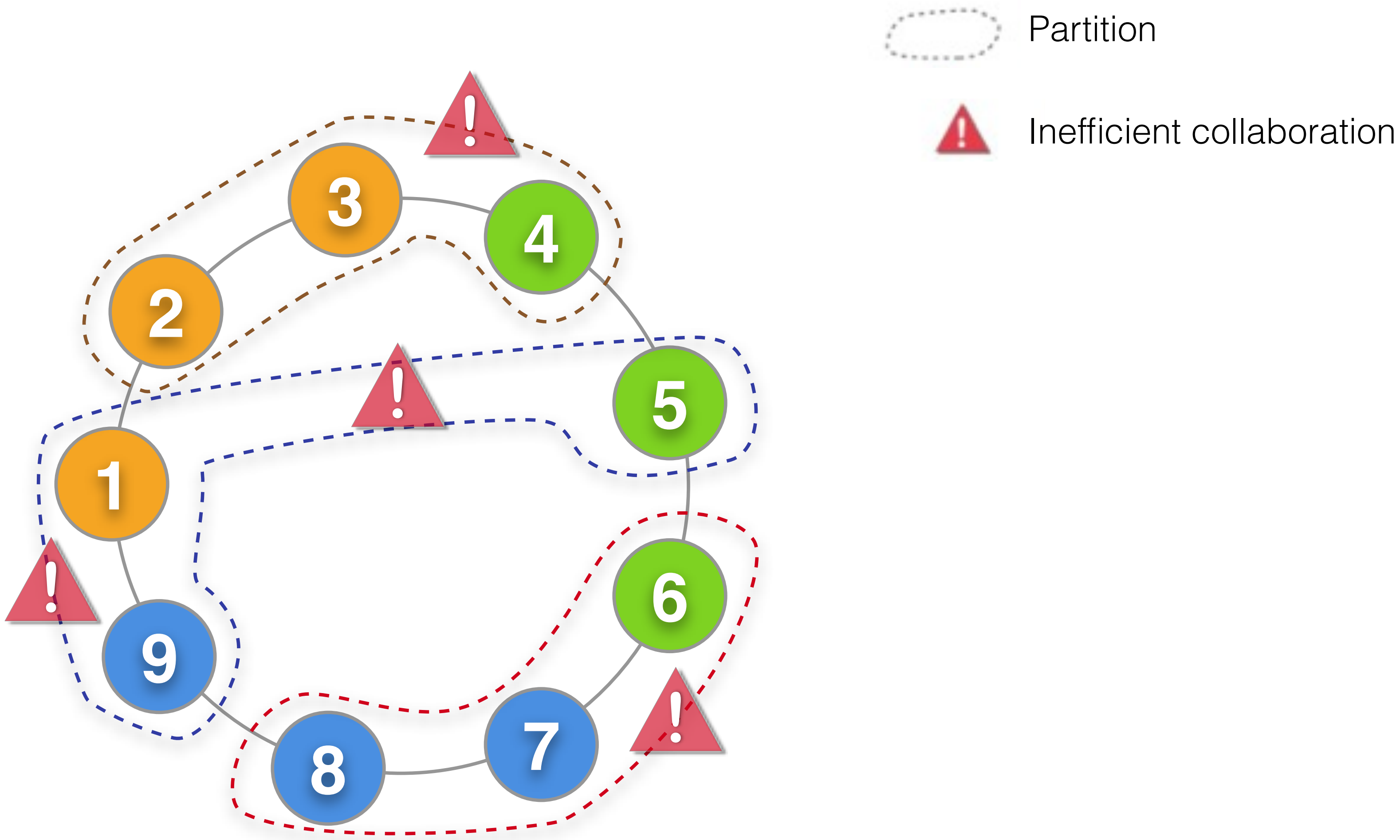


⬭ Partition

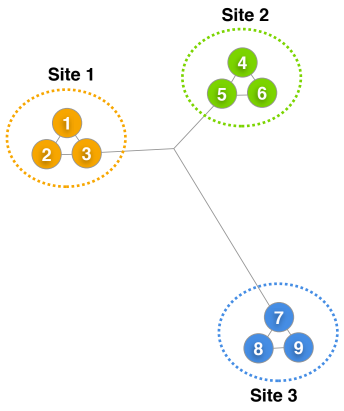
⚠ Inefficient collaboration

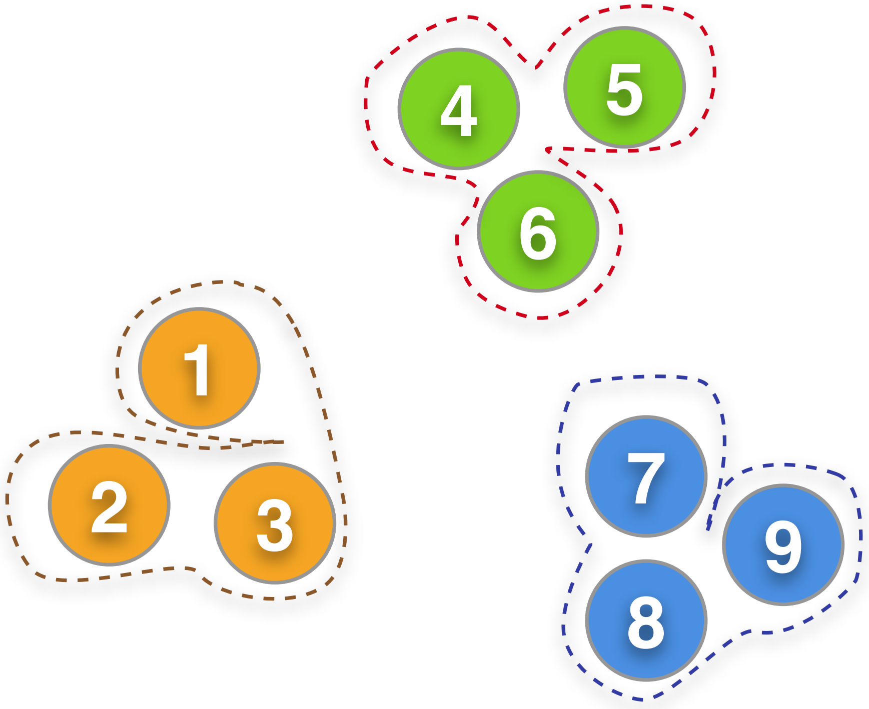
Chord cannot promote collaboration between close nodes





Close nodes should collaborate first

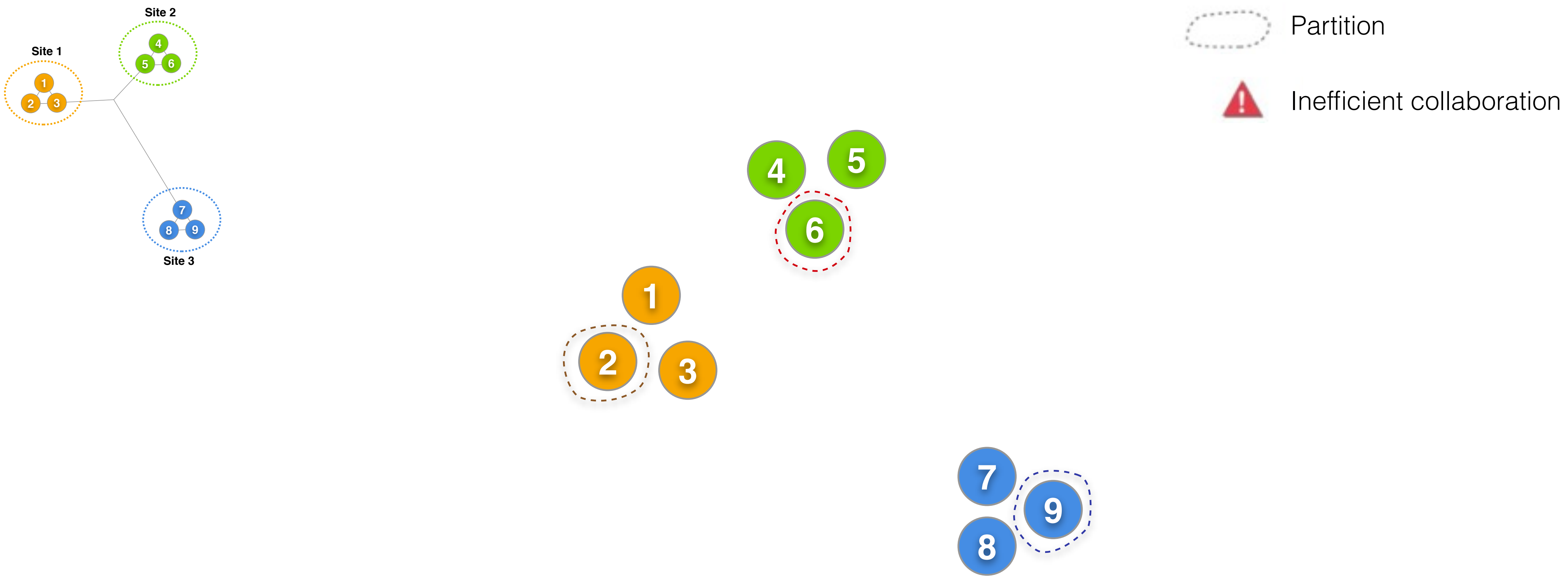




⬚ Partition

⚠ Inefficient collaboration

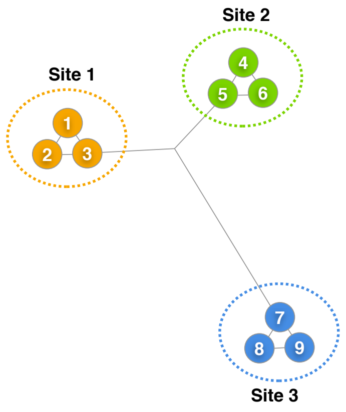
Close nodes should collaborate first



Partition

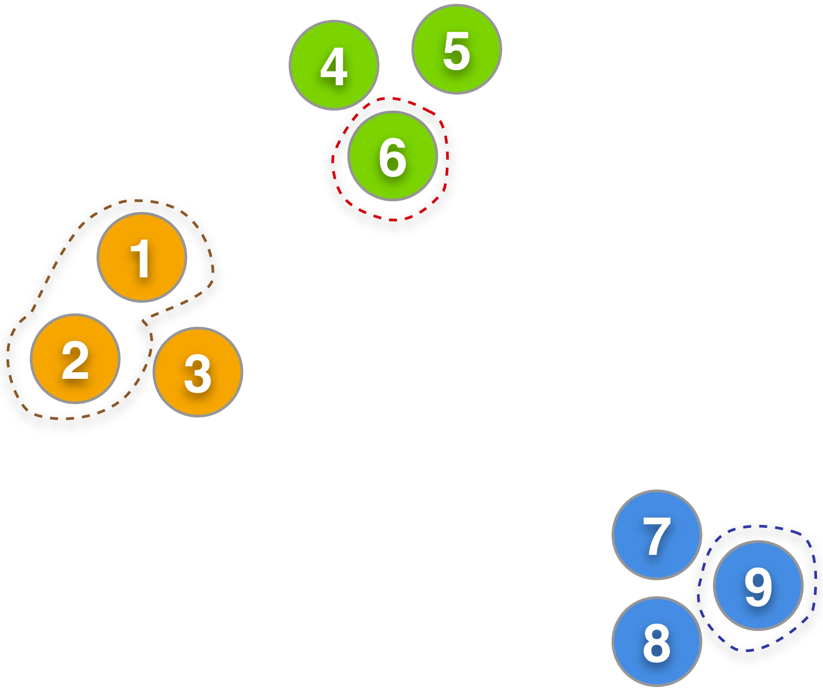
⚠ Inefficient collaboration

Close nodes should collaborate first

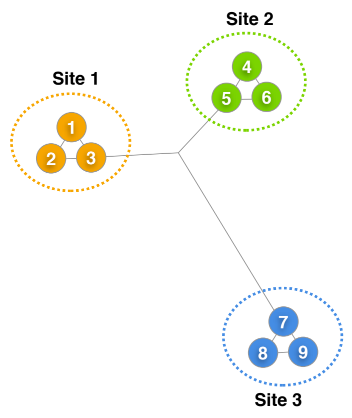


⬭ Partition

⚠ Inefficient collaboration

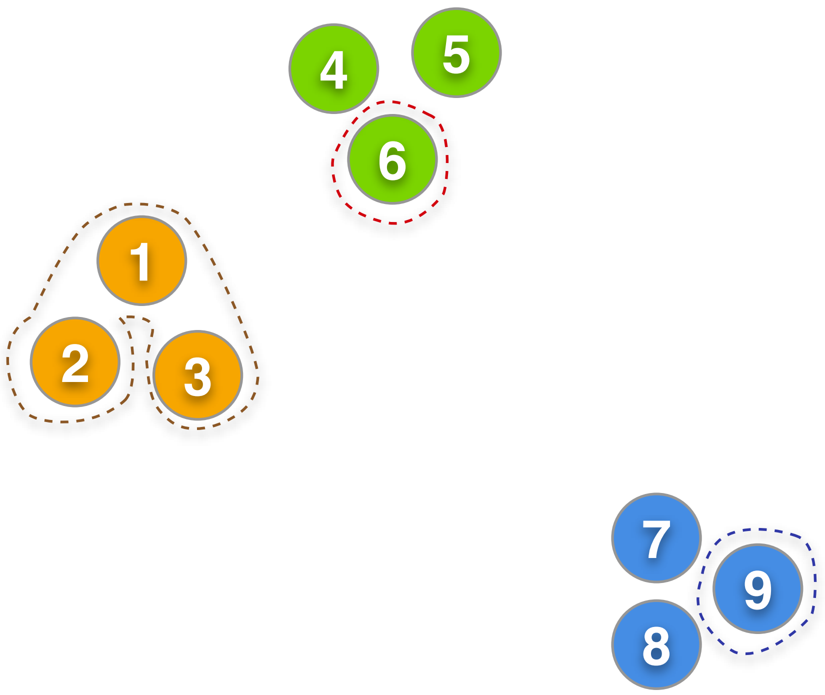


Close nodes should collaborate first

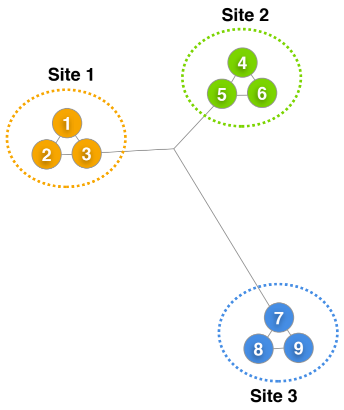


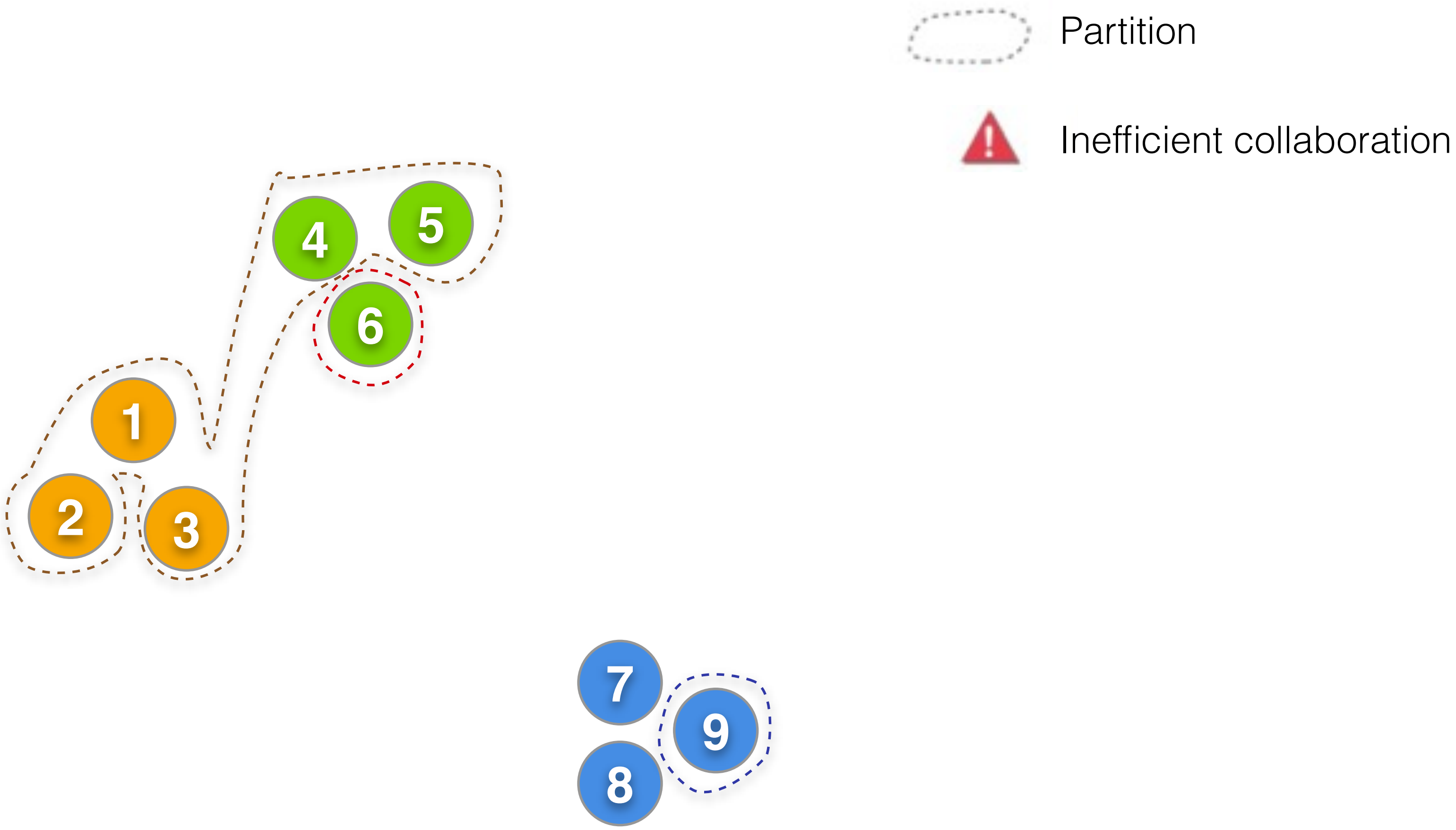
⬭ Partition

⚠ Inefficient collaboration

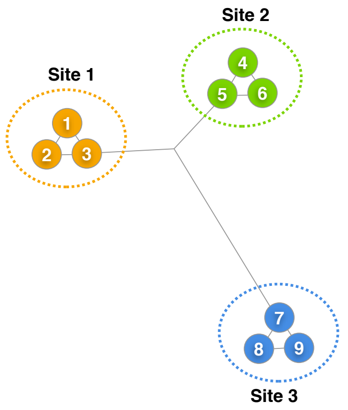


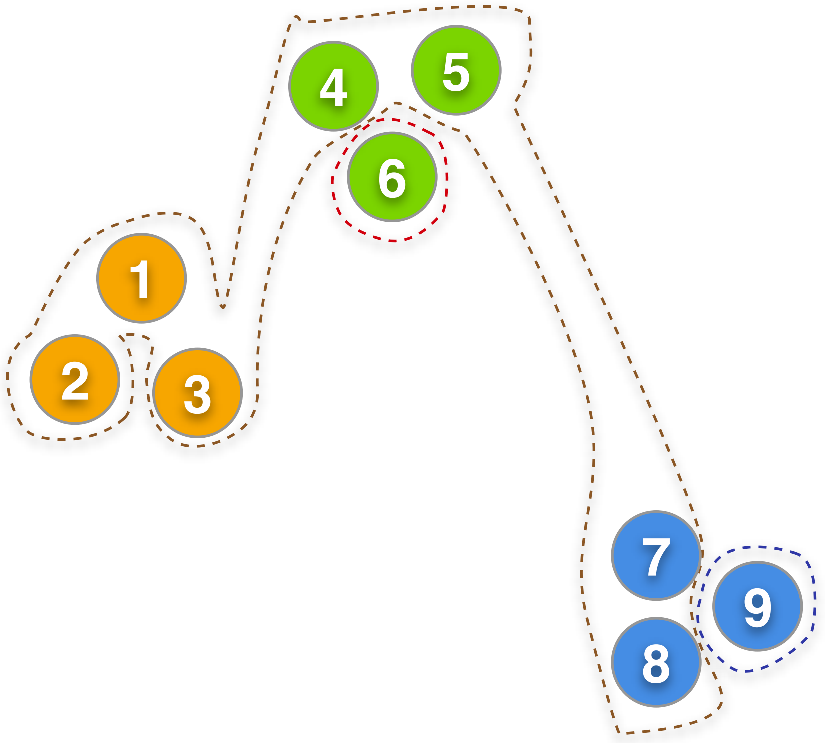
Close nodes should collaborate first





Close nodes should collaborate first

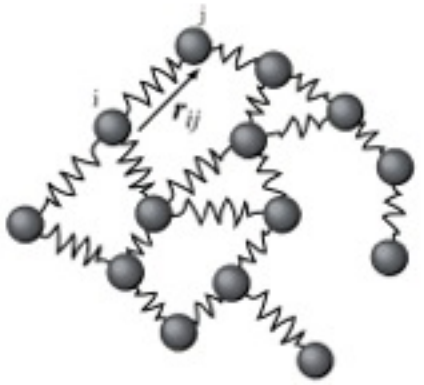




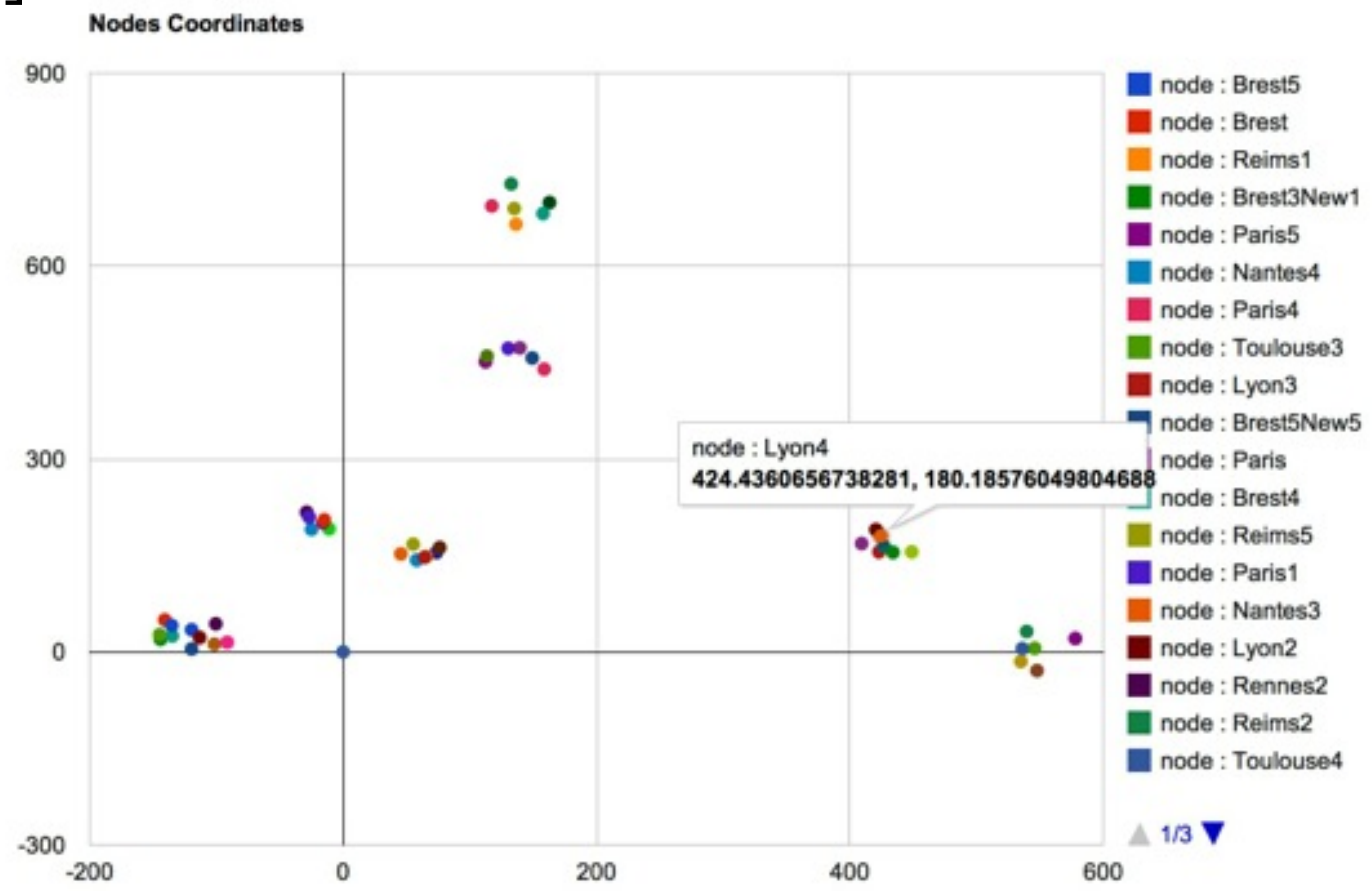
⬭ Partition

⚠ Inefficient collaboration

Vivaldi, a distributed coordinate system

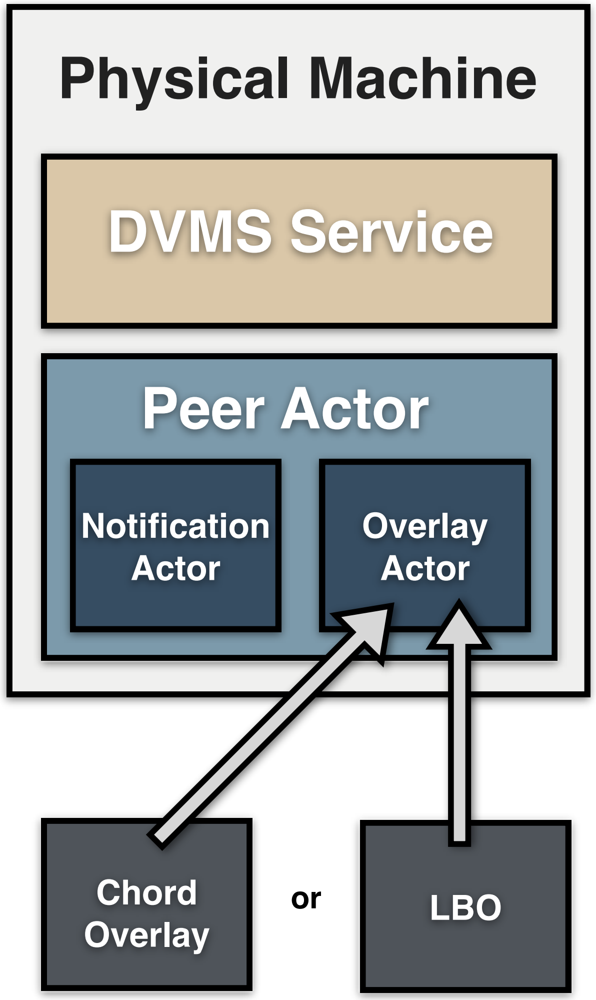


- Based on “Spring systems” [Dabek2004]
- Contacts are exchanged randomly between nodes
- Latency is measured => spring tension



Introduction of locality properties: the DVMS case

- DVMS uses the PeerActor model, where services leverage overlay networks.
- Development of a Locality Based Overlay (LBO).
- It uses the Vivaldi coordinate system.
- Through the use of the PeerActor architecture, DVMS collaborate with close neighbours to perform migrations.



Results

- The use of the LBO has increased the intra-site migration ratio

- Maximize intra-site migrations and by favouring cooperation between close nodes.

- Inter-sites collaborations have become more efficient (migration between close sites).

- **These results will be published at Europar2014 [3] and EIT ICT Labs [6].**

	Chord	***LBO***
Average	***0.496***	***0.863***
Minimum	***0.378***	***0.798***
Maximum	***0.629***	***0.935***

Comparison of intra-site migration ratio
protocol: 4 sites, 10 nodes per site
and number of VMs = 1,3 x number of core.

Large scale experiments

- Experiments discussed in the Europar article: a promising glimpse of using locality properties to improve collaboration between nodes, but not sufficient to validate the concept at large scale.

- ***Grid'5000 scale challenge [4]***: *launching an experiment that contains thousands of VMs (at least 5k) on several geographical sites (8)*.

- Collaboration with Laurent Pouilloux, from the AVALON research team.

- Leveraging *vm5k*, a tools that can deploy and configure thousands of VMs on grid'5000 [Imbert2013].

- Use *vm5k* since the first year: main beta tester.

- Results will be presented next week (hopefully :-)).
  Complete experiment facing a lot of possible issue (kadeploy, global kavlan, vm5k, VMs crash…)

Architecting the LUC-OS over OpenStack

LUC-OS

- Locality Based Utility Computing OS (LUC-OS):

- A **fully distributed** Cloud-OS that enables to use and operate a massively distributed infrastructure at WAN scale, leveraging **locality properties** in order to organise efficient **cooperations**.

- To address fault tolerance and energy concerns.

- To address the network overhead, micro/nano DCs will be located on ISP point of presence.

IaaS reference architecture [Moreno2012]

- Designing the LUC-OS is a complex tasks.
- Defined a reference architecture for IaaS managers.
- Using this architecture:
    - Minimize conception and implementation effort

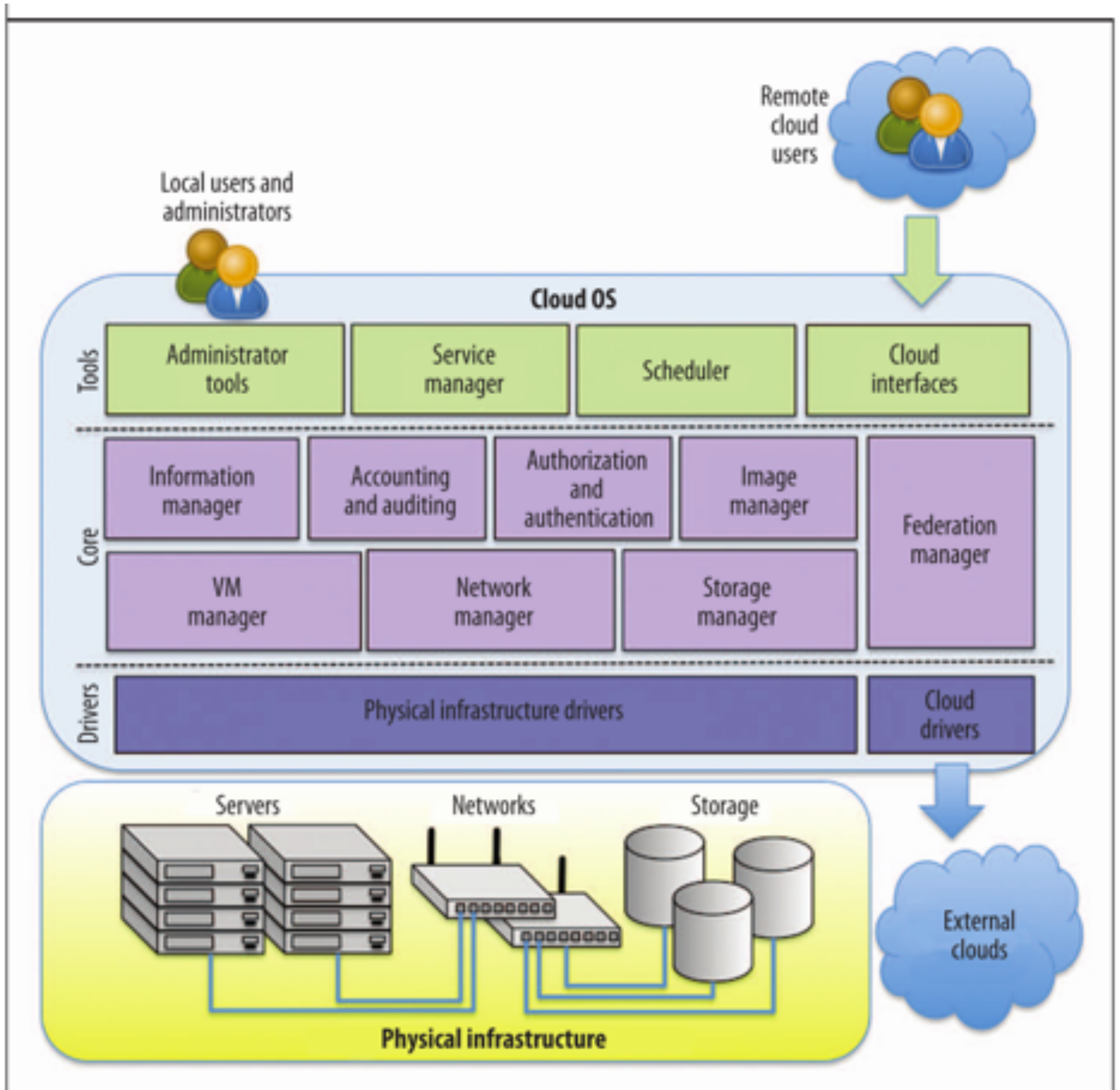
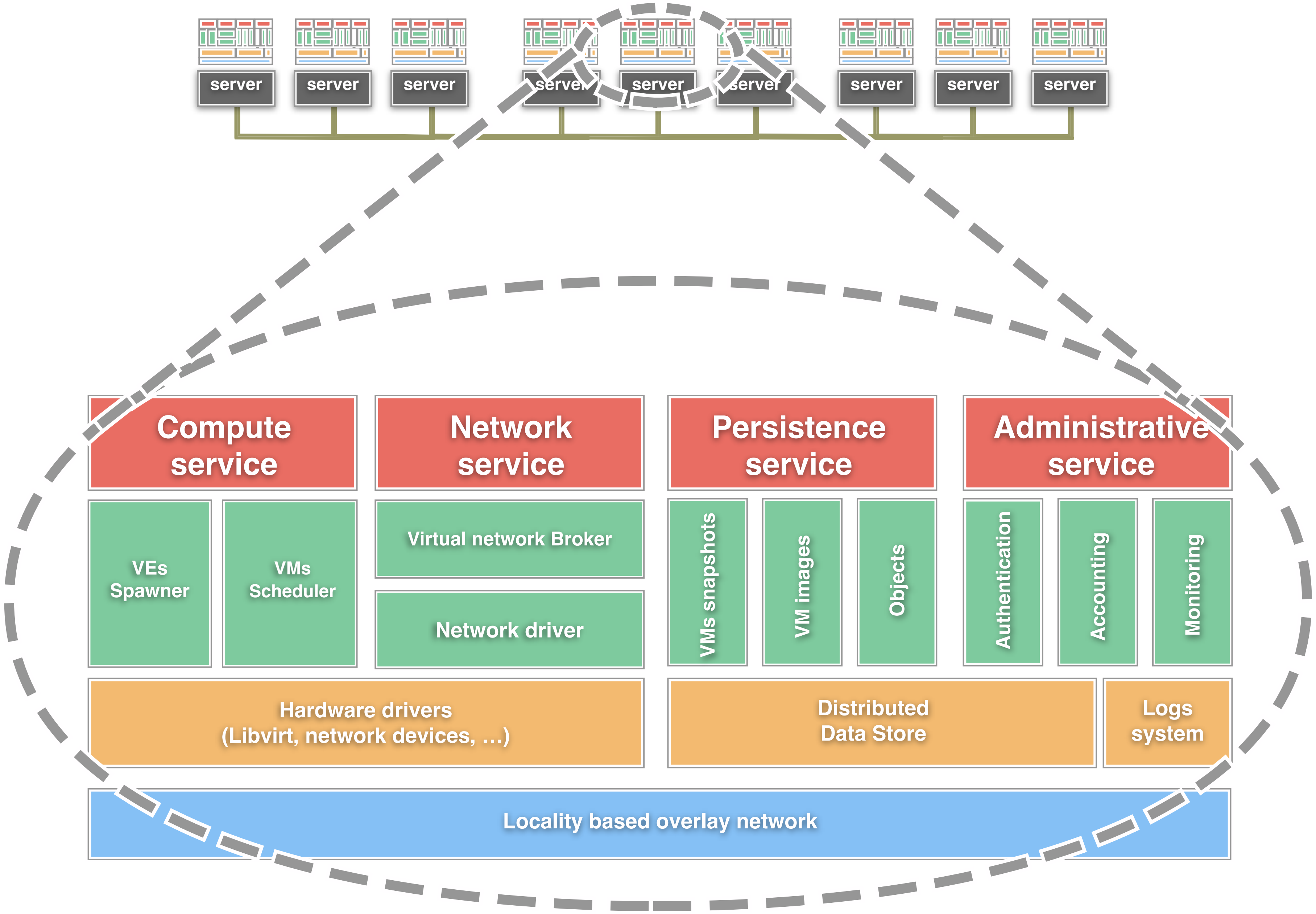


Figure 1. The cloud OS, the main component of an IaaS cloud architecture, is organized in three layers: drivers, core components, and high-level tools.

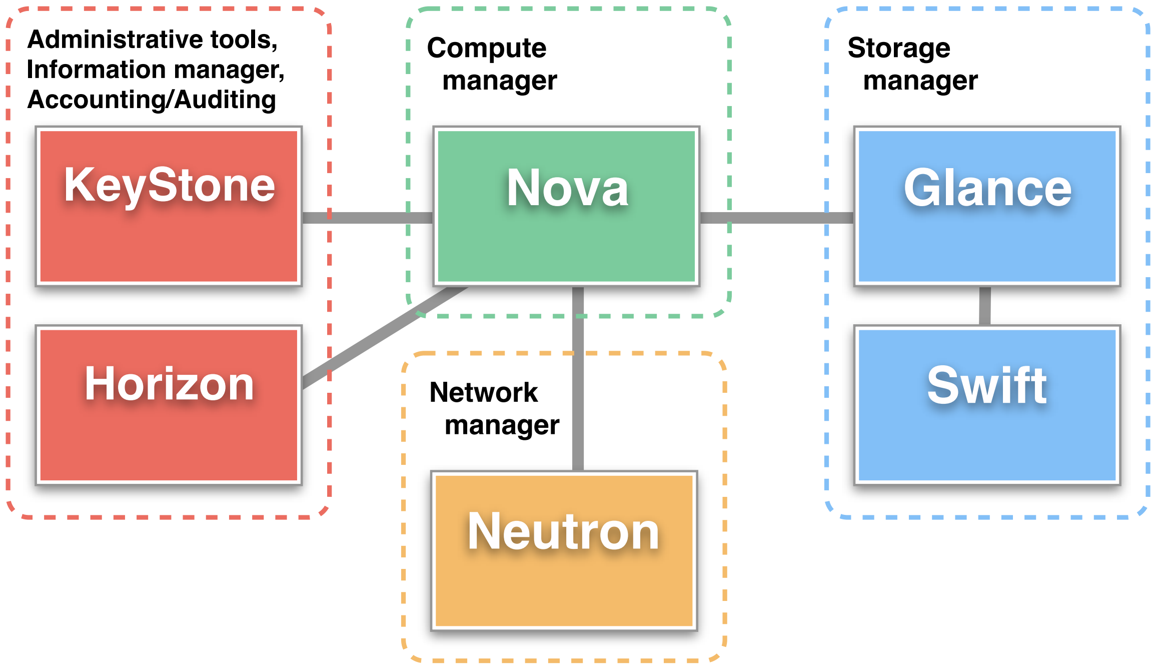
Figure extracted from [Moreno2012]

Moreno’s reference architecture revisited to fit with the LUC-OS



OpenStack

- Designing a Cloud-OS from scratch will be an herculean work: we propose to leverage existing mechanisms.
- OpenStack is an open source project that aims at developing a self sufficient IaaS manager.



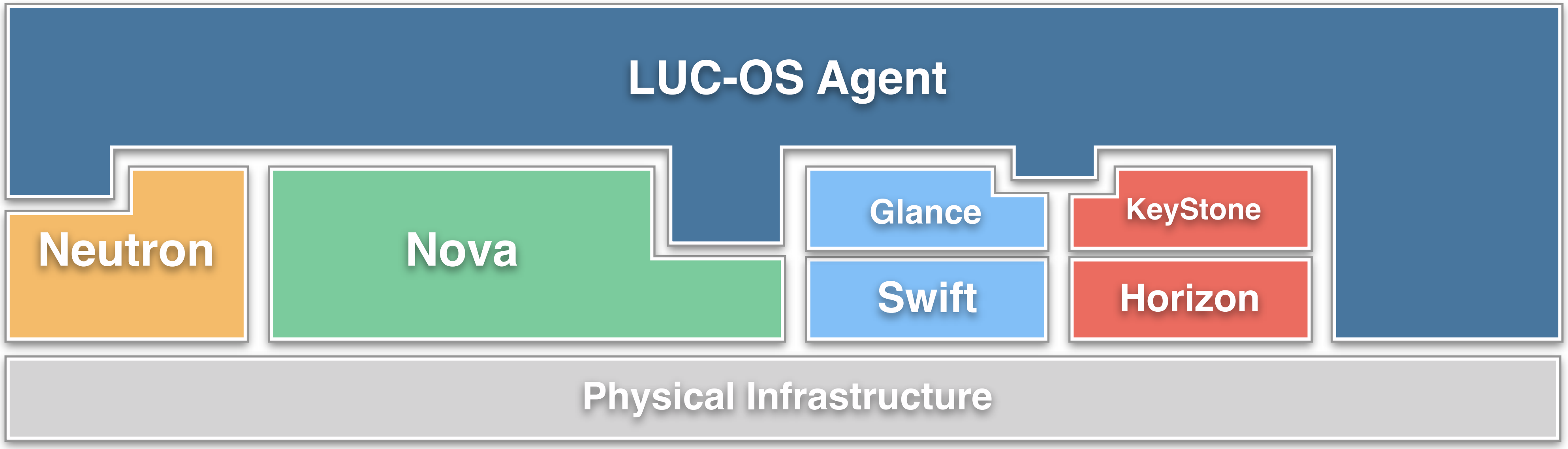
openstack™

Designing the LUC-OS on top of OpenStack

- The LUC-OS will rely on a multi-agent architecture.

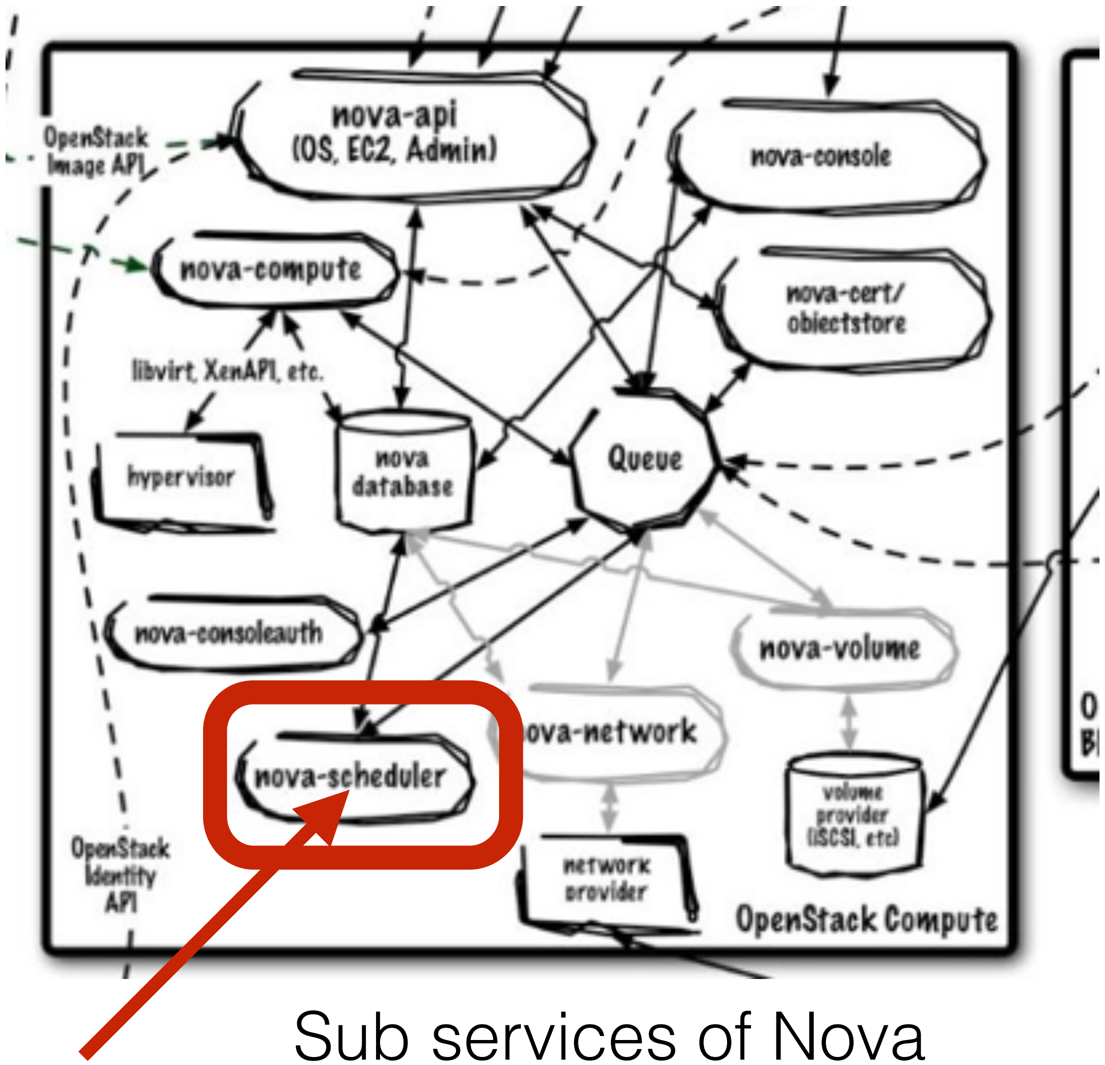
- Some services of the LUC-OS will entirely reuse implementation from OpenStack (***Swift***).

- Some services will “adapt” OpenStack to the LUC-OS (***Nova***).



Revisiting existing mechanisms

- Nova contains a scheduler (nova-scheduler).
- Replace nova-scheduler by a custom scheduler.
- Each incoming message will be forwarded to DVMS.
- Messages produced by DMVS will be translated and sent to other OpenStack services.



Sub services of Nova

Publications

Second year:

- [3] Jonathan Pastor, Marin Bertier, Frédéric Desprez, Adrien Lèbre, Flavien Quesnel, and Cédric Tedeschi. **Locality-aware Cooperation for VM Scheduling in Distributed Clouds**. In Euro-Par 2014, Porto, Portugal, August 2014.

- [2] Adrien Lèbre, Jonathan Pastor, Marin Bertier, Frédéric Desprez, Jonathan Rouzaud-Cornabas, Cédric Tedeschi, Paolo Anedda, Gianluigi Zanetti, Ramon Nou, Toni Cortes, Etienne Rivière, and Thomas Ropars. **Beyond The Cloud, How Should Next Generation Utility Computing Infrastructures Be Designed?**. Research Report RR-8348, INRIA, July 2013, to appear in Springer Book "Cloud computing - Challenges, Limitations and R&D solutions".

First year:

- [1] Flavien Quesnel, Adrien Lèbre, Jonathan Pastor, Mario Südholt, and Daniel Balouek. **Advanced Validation of the DVMS Approach to Fully Distributed VM Scheduling**. In ISPA' 13: The 11th IEEE International Symposium on Parallel and Distributed Processing with Applications, Melbourne, Australia, July 2013.

Dissemination

Second year:

- [4] Jonathan Pastor, Laurent Pouilloux. **VM5k and DVMS Deploying and Managing Thousands of Virtual Machines on Hundreds of Nodes Distributed Geographically**. Grid'5000 spring school, Lyon, France, June 2014.

- [5] Jonathan Pastor, Adrien Lèbre, Frédéric Desprez. **Designing a massively distributed IaaS toolkit by revisiting OpenStack internals**. VHPC 2014, Porto, Portugal, August 2014, <u>currently under review</u>.

- [6] Jonathan Pastor. **VM scheduling for Capacity Planning in Distributed Clouds**. Poster, EIT ICT labs, Cloud Computing symposium.

Teaching activities

- Introduction to web programming:
  lecture, practical session, organization (13h)

- New generation languages (Javascript, Scala):
  tutorial, practical session (17h)

- Programming methodology (Java, Data structure):
  tutorial, practical session (35h)

Conclusion

What have been done

- Introduction of locality properties
  - Integration in DVMS.
  - Large scale experiments (grid'5000 challenge).
- First software architecture of the LUC-OS, leveraging OpenStack.

Ongoing and Future work

- Build a first prototype of the LUC-OS over OpenStack (primary objective).
- Define software programming rules to make the LUC-OS development easier(secondary objective).
  - Through the use of advanced programming abstraction (functional programming: promise/future and Monads)
  - On going use case: DVMS.

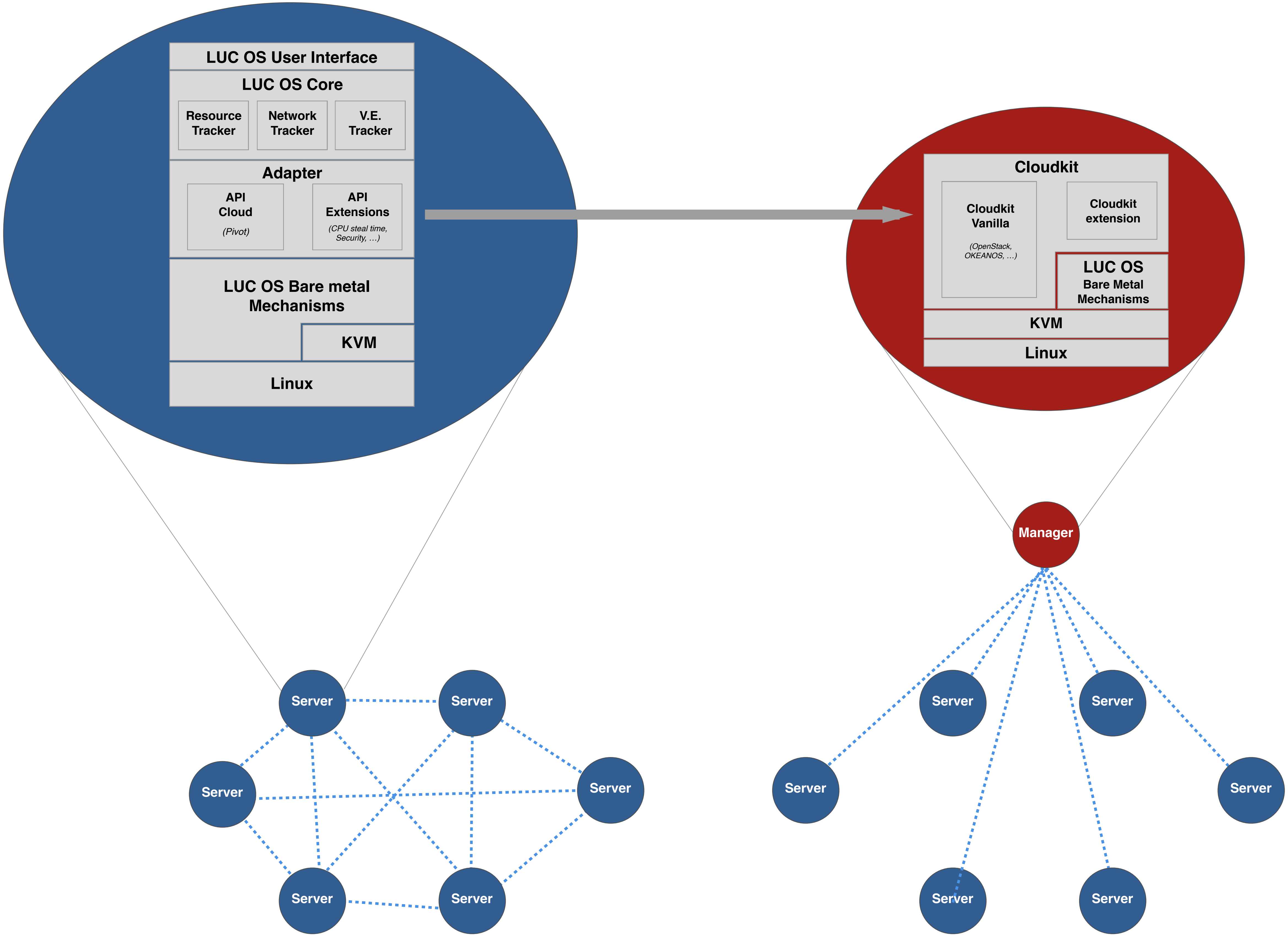
Bibliography

- [Dabek2004] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. **Vivaldi: A decentralized network coordinate system**. In ACM SIGCOMM Computer Communication Review, volume 34, pages 15–26. ACM, 2004.

- [Greenberg2009] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. **The cost of a cloud: research problems in data center networks**. ACM SIGCOMM Computer Communication, Review, 39(1):68–73, 2008.

- [Moreno2012] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente. **Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures**. Computer, 45(12):65–72, 2012.

- [IEEE2012] I. . E. W. Group. **IEEE 802.3TM Industry Connections Ethernet Bandwidth, Assessment, July 2012**.



LUC OS User Interface

LUC OS Core

Resource Tracker

Network Tracker

V.E. Tracker

Adapter

API Cloud

(Pivot)

API Extensions

(CPU steal time, Security, …)

LUC OS Bare metal Mechanisms

KVM

Linux

Cloudkit

Cloudkit Vanilla

(OpenStack, OKEANOS, …)

Cloudkit extension

LUC OS

Bare Metal Mechanisms

KVM

Linux

Manager

Server

Server

Server

Server

Server

Server

Server

Server

Server

Server

Server

Server