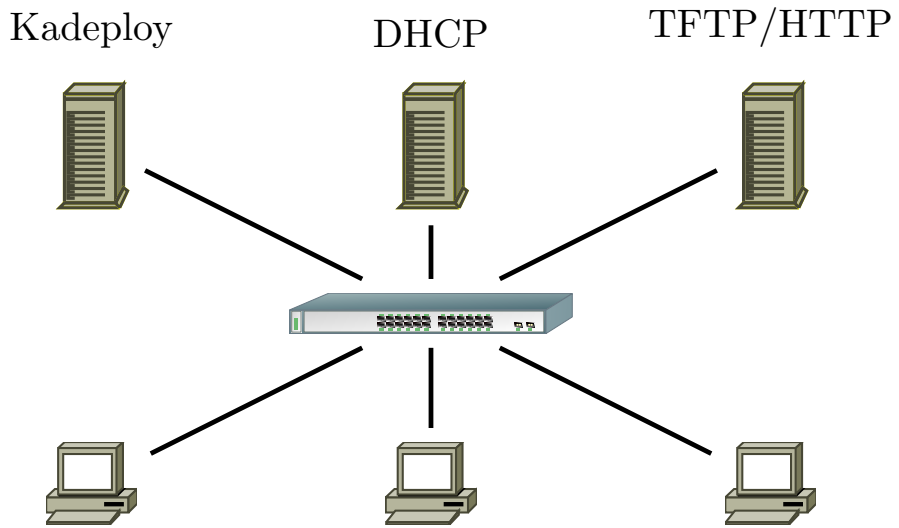
Deploying large-scale virtual infrastructures with Kadeploy3

Luc Sarzyniec, Sébastien Badia, Emmanuel Jeanvoine, Lucas Nussbaum

Plan

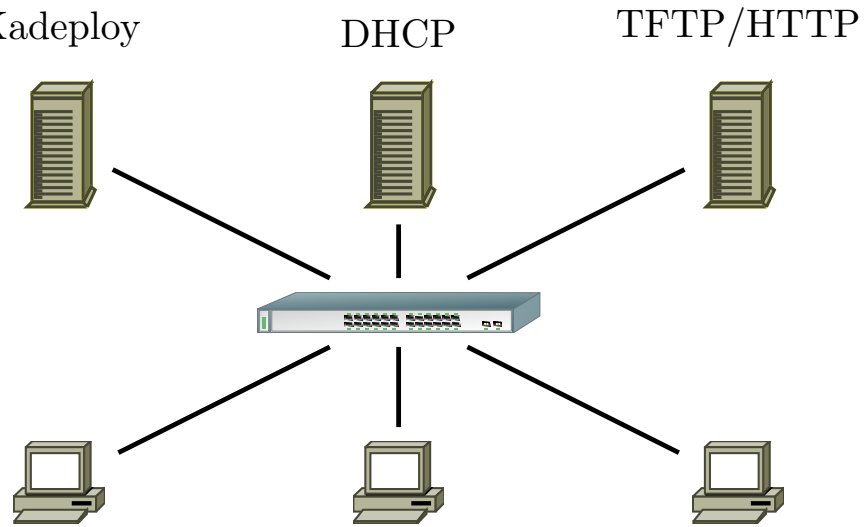
Kadeploy3



- Used by Grid'5000 users to install/reinstall compute nodes
- Designed for scalability
- Support of a broad range of systems (Linux, Xen, *BSD, etc.)
- Manages catalog of images and user permissions
- Built on top of PXE, DHCP, TFTP (or HTTP)

Kabootstrap

Install your own Kadeploy3 testbed on cluster nodes



- Fully automated process
- Automatically gather network informations
- Get hardware information using Grid'5000 API
- Install and configure each service of the Kadeploy3 ecosystem knowing the current network/hardware configuration
- Kadeploy3 is network-intrusive
  - Usage of VLANs

Our experiment

Goal

Evaluate the scalability of Kadeploy3

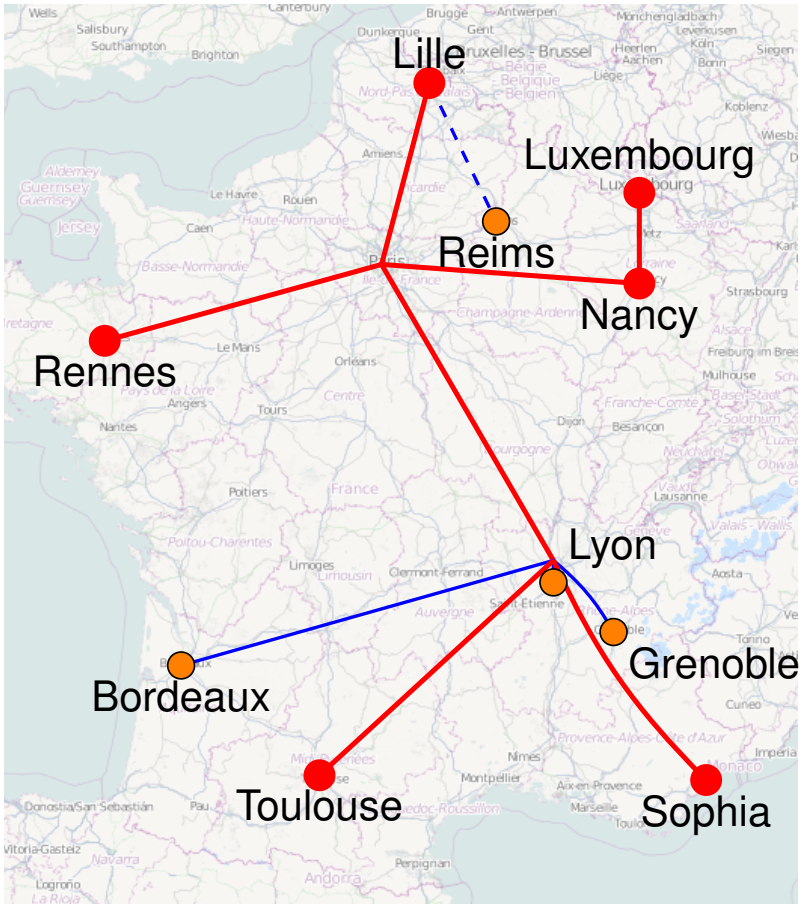
Problem

Not enough machines on Grid'5000

Solution

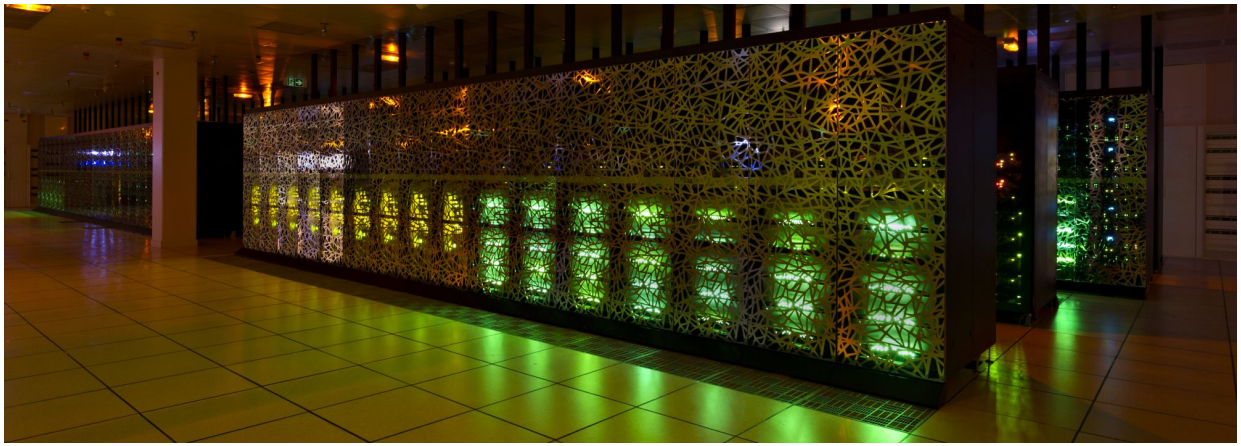
Deployment of virtual machines

- Boot over Network
- Execution of parallel commands
- CPU intensive tasks
- File broadcast



Plan

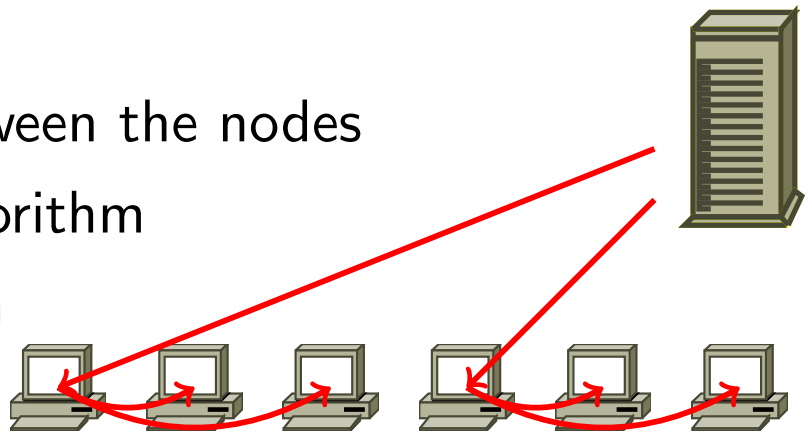
Scalability challenges in Kadeploy



- Reboot commands
  - Windowed operations
- Boot over network
  - Use HTTP instead of TFTP (iPXE)
- Parallel command execution
  - Hierarchical connections, TakTuk
- Environment image broadcast
  - Topology-aware chain, Kastafior

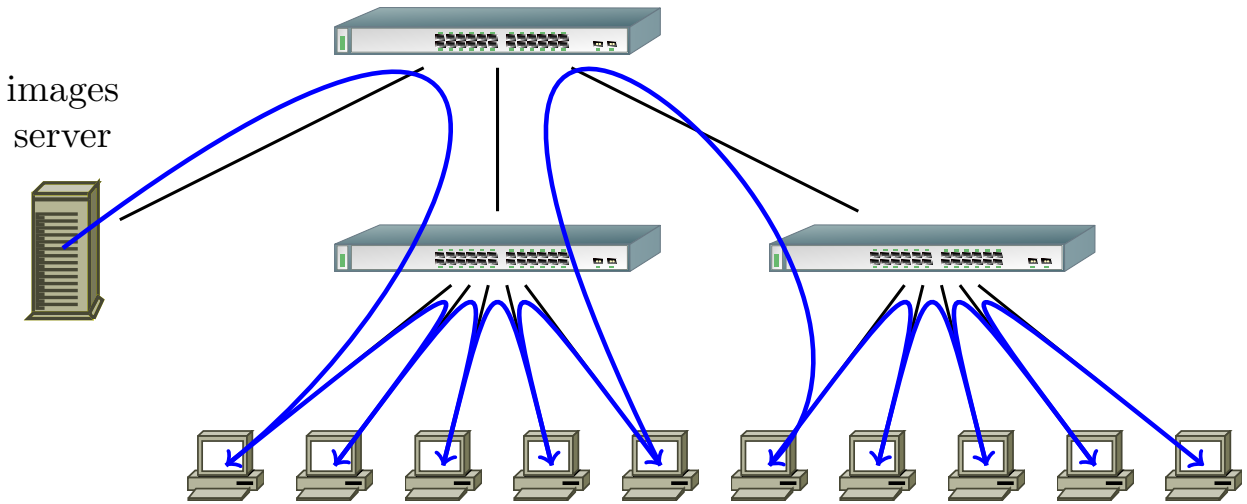
TakTuk (G. Huard, LIG Grenoble)

- Hierarchical connections between the nodes
- Adaptative work-stealing algorithm
- Auto-propagation mechanism
- Installed on Grid'5000
- Recommended by Grid'5000 team



`http://taktuk.gforge.inria.fr/`

TakTuk, Adaptive Deployment of Remote Executions,
`B. Claudel, G. Huard and O. Richard`, *HPDC'2009*



Topology aware chained file broadcast

- Chain-based broadcast
- Initialization of the chain with tree-based parallel command
- Saturation of full-duplex networks in both directions

Plan

Experimental process

1. Virtual testbed preparation
   - 1.1 Reserve and reinstall all nodes on Grid'5000 $\rightsquigarrow$ 20m
     - Usage of Grid'5000 API
   - 1.2 Prepare Service and Host nodes $\rightsquigarrow$ 5m
     - Pre-configure Host and Service machines (network interfaces)
     - Guess how much VMs each Host can run
     - Dispatch VMs in sub-networks (one per Grid'5000 site)
     - Launch the VMs on each Host
   - 1.3 Install and configure Service nodes with Kabootstrap $\rightsquigarrow$ 15m
2. One or more Kadeploy runs

   Login to the 'kabootstrapped' frontend and start deployments

Experimental process

1. Virtual testbed preparation
   - 1.1 Reserve and reinstall all nodes on Grid'5000 $\rightsquigarrow$ 20m
     - Usage of Grid'5000 API
   - 1.2 Prepare Service and Host nodes $\rightsquigarrow$ 5m
     - Pre-configure Host and Service machines (network interfaces)
     - Guess how much VMs each Host can run
     - Dispatch VMs in sub-networks (one per Grid'5000 site)
     - Launch the VMs on each Host
   - 1.3 Install and configure Service nodes with Kabootstrap $\rightsquigarrow$ 15m
2. One or more Kadeploy runs

   Login to the 'kabootstrapped' frontend and start deployments

Experimental process

1. Virtual testbed preparation
   - 1.1 Reserve and reinstall all nodes on Grid'5000 $\rightsquigarrow$ 20m
     - Usage of Grid'5000 API
   - 1.2 Prepare Service and Host nodes $\rightsquigarrow$ 5m
     - Pre-configure Host and Service machines (network interfaces)
     - Guess how much VMs each Host can run
     - Dispatch VMs in sub-networks (one per Grid'5000 site)
     - Launch the VMs on each Host
   - 1.3 Install and configure Service nodes with Kabootstrap $\rightsquigarrow$ 15m
2. One or more Kadeploy runs

   Login to the 'kabootstrapped' frontend and start deployments

Experimental process

1. Virtual testbed preparation
   - 1.1 Reserve and reinstall all nodes on Grid'5000 $\rightsquigarrow$ 20m
     - Usage of Grid'5000 API
   - 1.2 Prepare Service and Host nodes $\rightsquigarrow$ 5m
     - Pre-configure Host and Service machines (network interfaces)
     - Guess how much VMs each Host can run
     - Dispatch VMs in sub-networks (one per Grid'5000 site)
     - Launch the VMs on each Host
   - 1.3 Install and configure Service nodes with Kabootstrap $\rightsquigarrow$ 15m
2. One or more Kadeploy runs

   Login to the 'kabootstrapped' frontend and start deployments

Results of a run on Grid'5000

Virtualized infrastructure

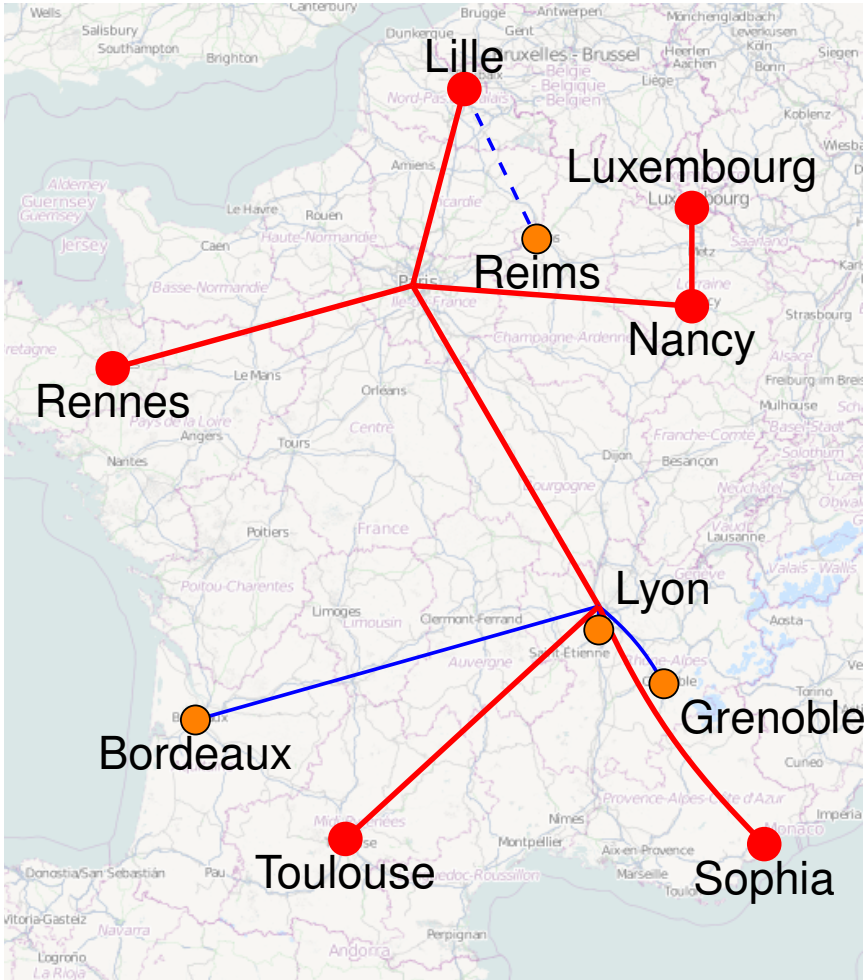
- 4552 VMs, 355 Hosts (≈2400c)
- 47 Service nodes
- 6 Grid'5000 sites

Virtual machines

- 2 VMs per core
- 914MB RAM per VM
  - 2-16 VMs per node

Deployment results

- 430MB environment
- Nodeset of $>$ 4000 nodes
- 58 minutes of deployment
- 4537 nodes deployed successfully (99.6%)



Limits to scalability

Nodes reboot

- Uses unreliable protocols: DHCP, TFTP
- Experimental workarounds
  - TFTP replaced by HTTP thanks to iPXE
  - VMs hard disks on ramdisk
  - Kadeploy3 tuning (reboot windows, big timeouts)

Remote command execution and broadcast of system image

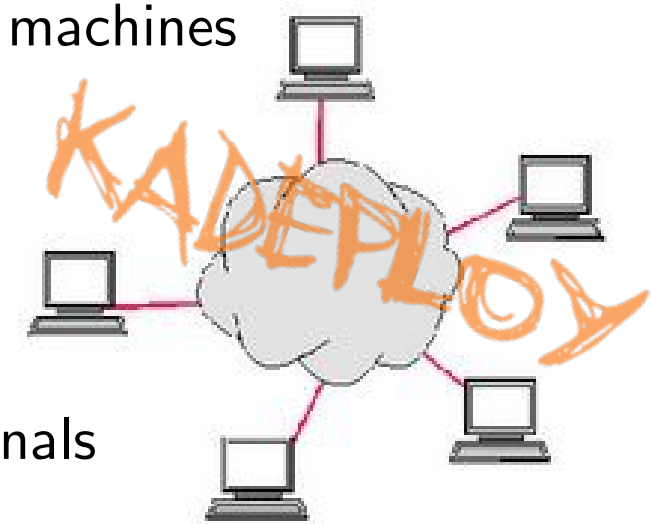
- Heavily stresses the network $\rightsquigarrow$ ARP, UDP and TCP timeouts
- Experimental workarounds
  - Custom iPXE ISO image with big timeouts
  - DNS settings (sub-networks architecture)
  - ARP tables size (on each Service node)

Demo

Conclusion

- Extensive use of Grid'5000 infrastructure
  - No specific privileges
  - Grid'5000 API
  - Kadeploy3 multi-site
  - Kavlan with global VLAN

- Configuration of a *Cloud* of KVM virtual machines
  - 4552 virtual machines
  - On 355 physical machines
  - From 6 sites of the Grid'5000 testbed



- Successfully deployed $>$ 4500 nodes
- A lot of improvements in Kadeploy3 internals

Special thanks to the Grid'5000 team that was really reactive when fixing the numerous bugs we found about KaVLAN and other pieces of the infrastructure ;)