Scalability Testing of Kadeploy using Virtual Machines on Grid'5000

Luc Sarzyniec, Sébastien Badia, Emmanuel Jeanvoine, Lucas Nussbaum

Scalability Testing of Kadeploy using Virtual Machines on Grid'5000

Luc Sarzyniec, Sébastien Badia, Emmanuel Jeanvoine, Lucas Nussbaum

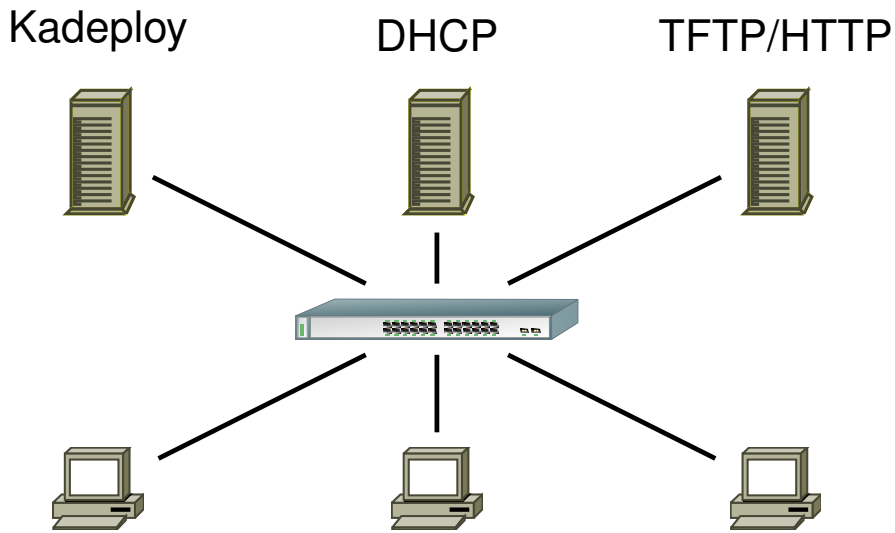
Grid'5000

Kadeploy – OS provisioning for clusters

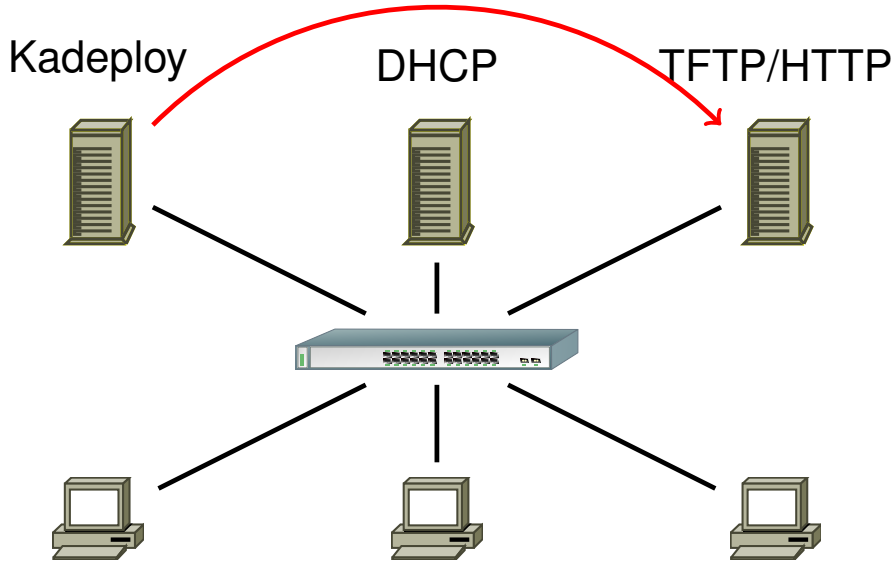
- Used by sysadmins to install/reinstall compute nodes
- Designed for scalability
  - That matters: faster reinstallation $\rightsquigarrow$ shorter downtime
- Built on top of PXE, DHCP, TFTP (or HTTP)
- Support of a broad range of systems (Linux, Xen, *BSD, etc.)
- Manages catalog of images and user permissions
- Open Source (GPL)

http://kadeploy3.gforge.inria.fr/

Process overview

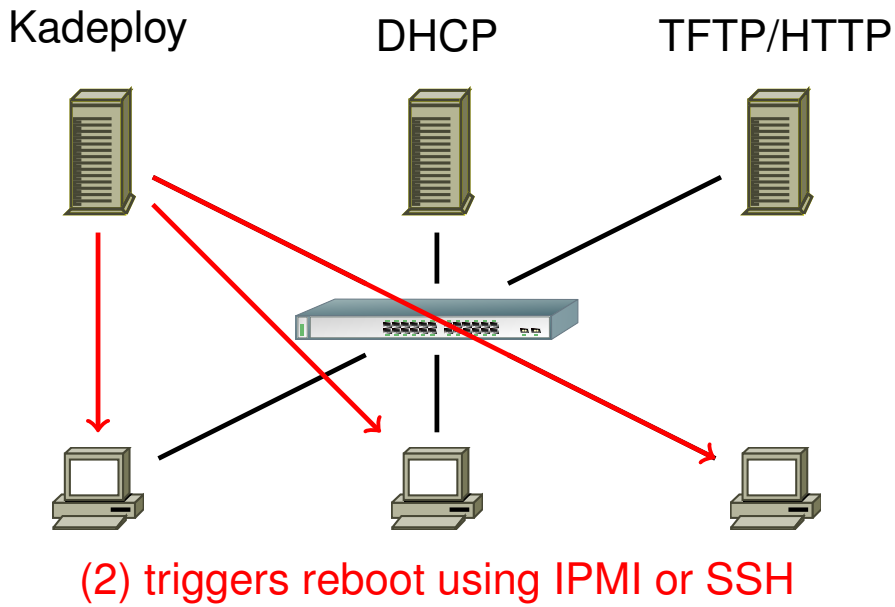


Process overview



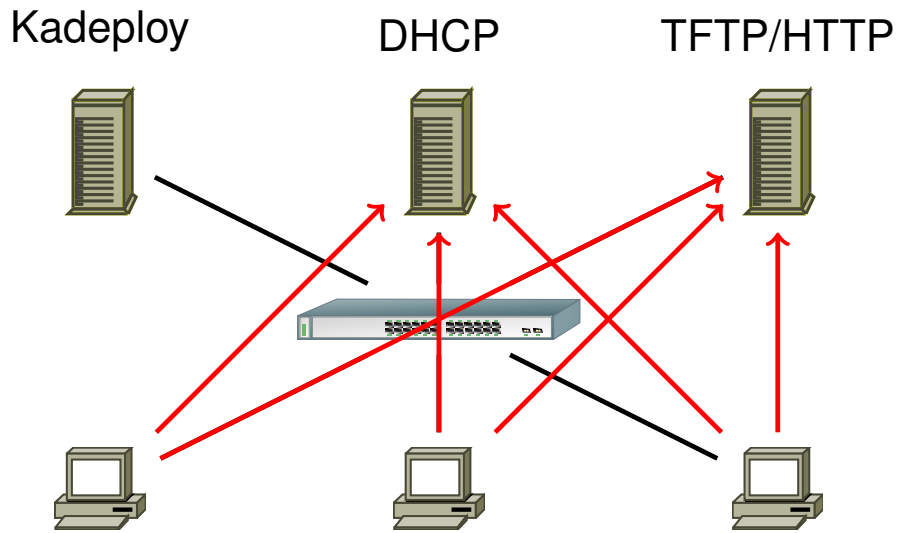
1. Kadeploy configures PXE profiles

Process overview



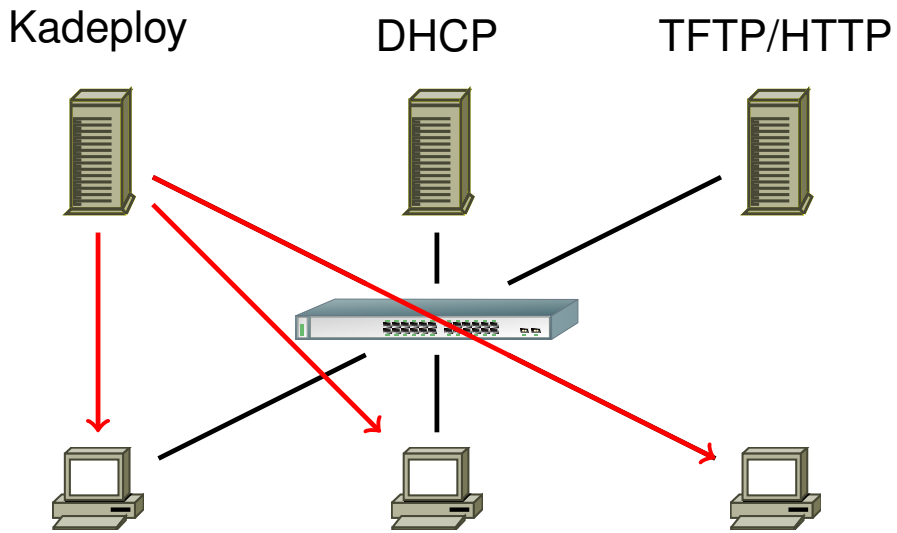
1. Kadeploy configures PXE profiles
2. Kadeploy triggers reboot using IPMI or SSH

Process overview



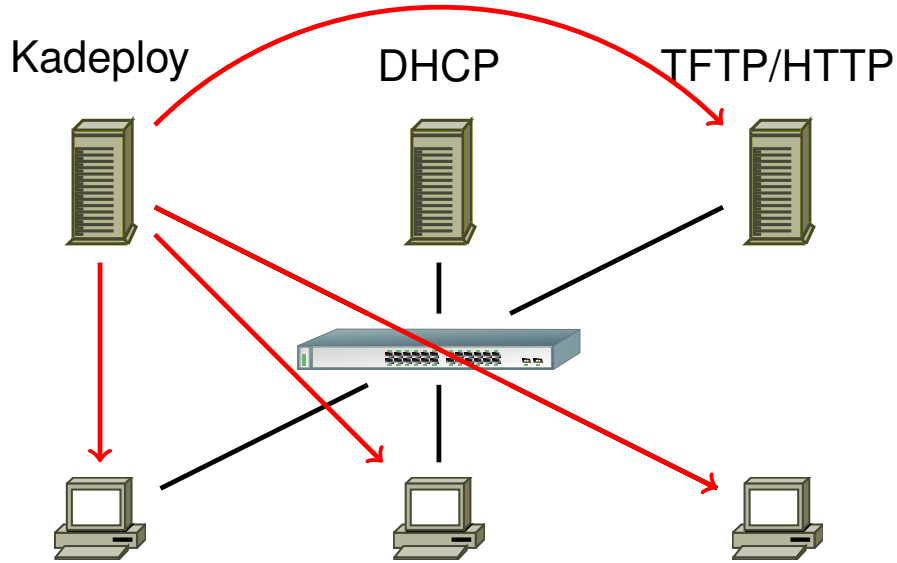
1. Kadeploy configures PXE profiles
2. Kadeploy triggers reboot using IPMI or SSH
3. Nodes boot to minimal deployment system sent over the network

Process overview



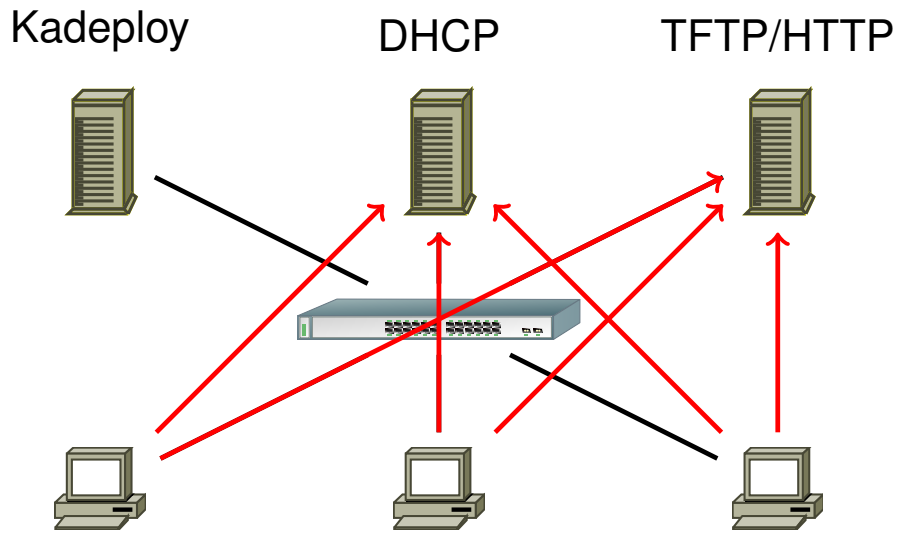
1. Kadeploy configures PXE profiles
2. Kadeploy triggers reboot using IPMI or SSH
3. Nodes boot to minimal deployment system sent over the network
4. Kadeploy configures nodes and sends system image

Process overview



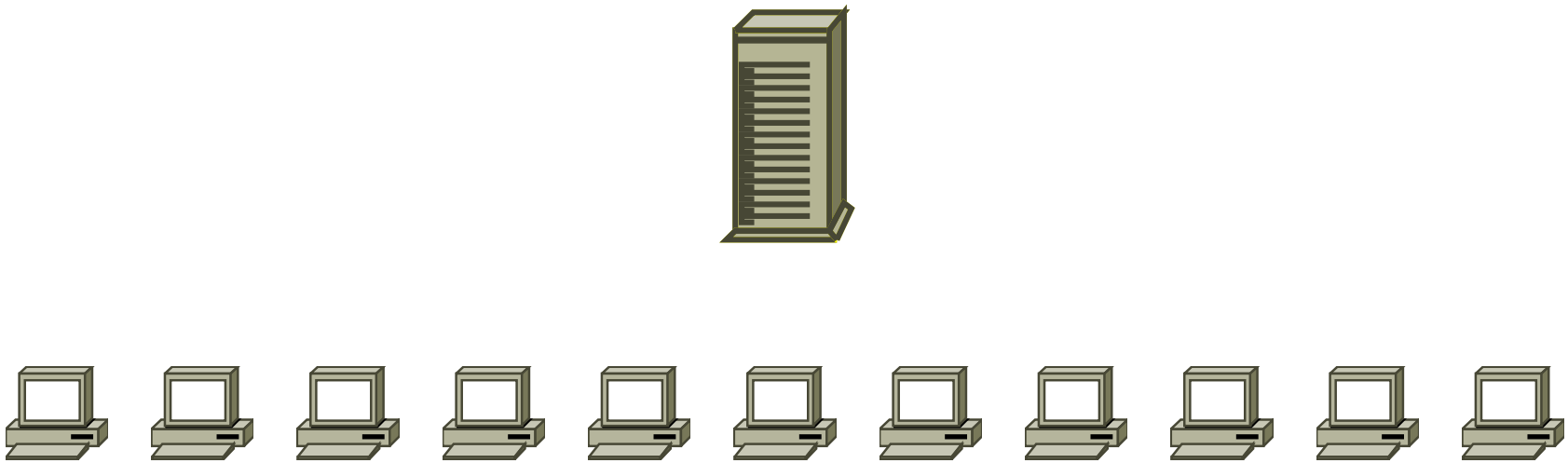
1. Kadeploy configures PXE profiles
2. Kadeploy triggers reboot using IPMI or SSH
3. Nodes boot to minimal deployment system sent over the network
4. Kadeploy configures nodes and sends system image
5. Kadeploy configures PXE profiles again and triggers reboot

Process overview

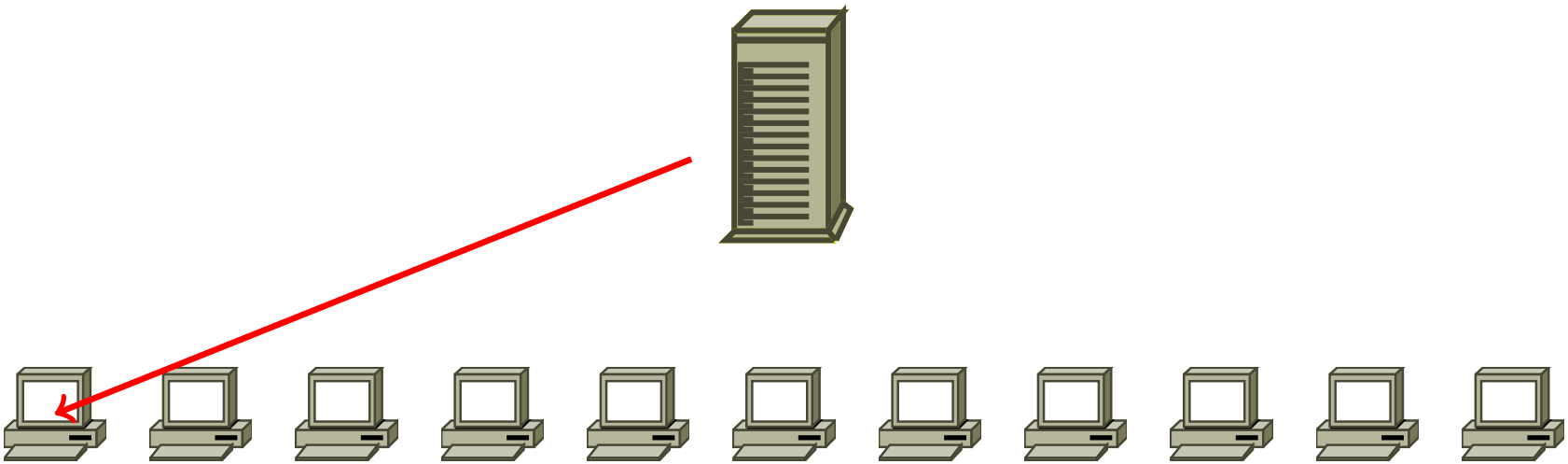


1. Kadeploy configures PXE profiles
2. Kadeploy triggers reboot using IPMI or SSH
3. Nodes boot to minimal deployment system sent over the network
4. Kadeploy configures nodes and sends system image
5. Kadeploy configures PXE profiles again and triggers reboot
6. Nodes boot to newly installed system

Scalable remote command execution with Taktuk

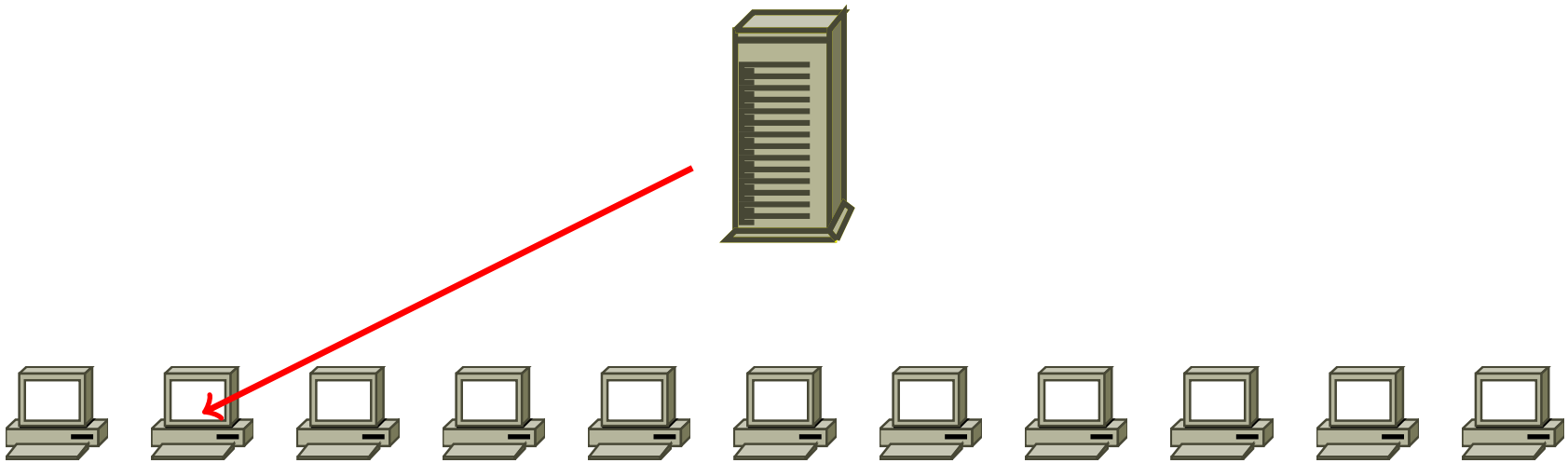


Scalable remote command execution with Taktuk



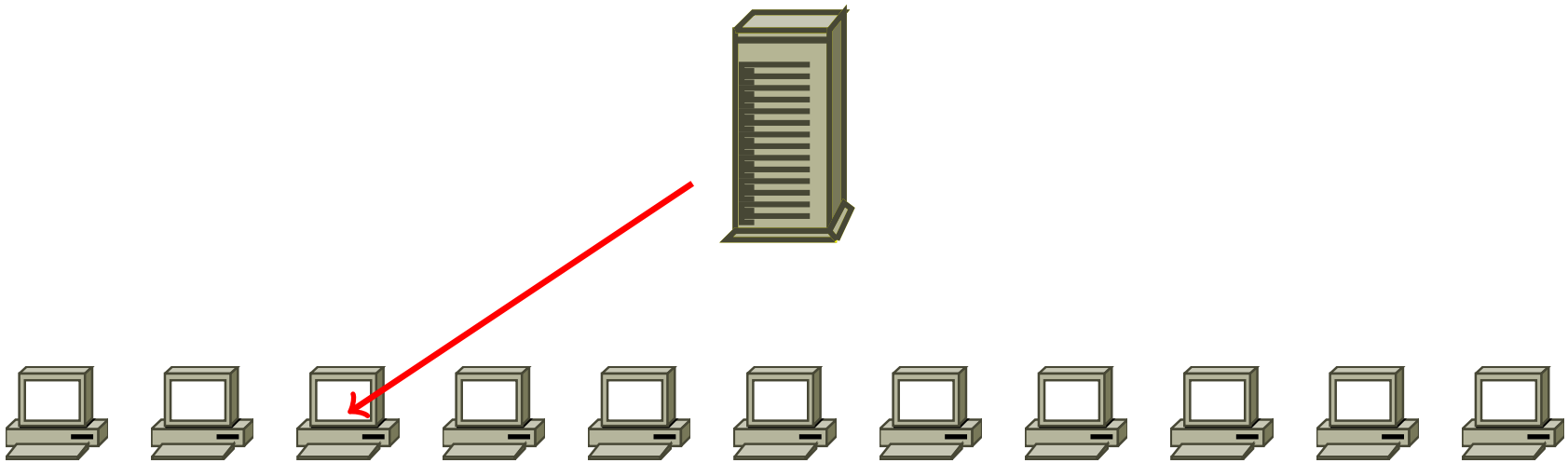
Sequential?

Scalable remote command execution with Taktuk



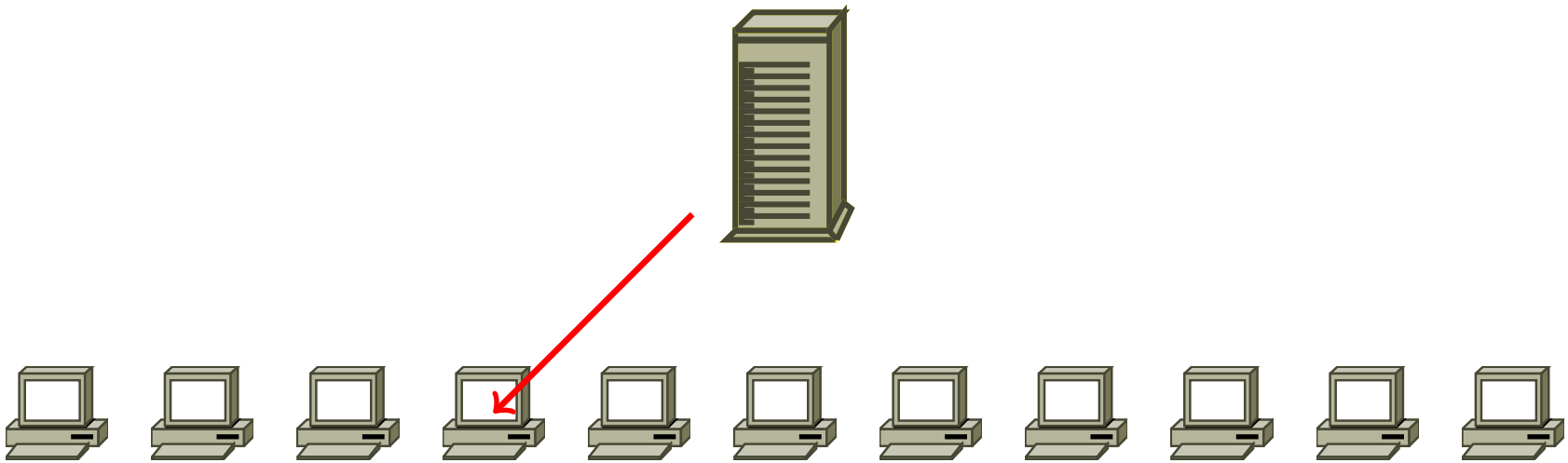
Sequential?

Scalable remote command execution with Taktuk



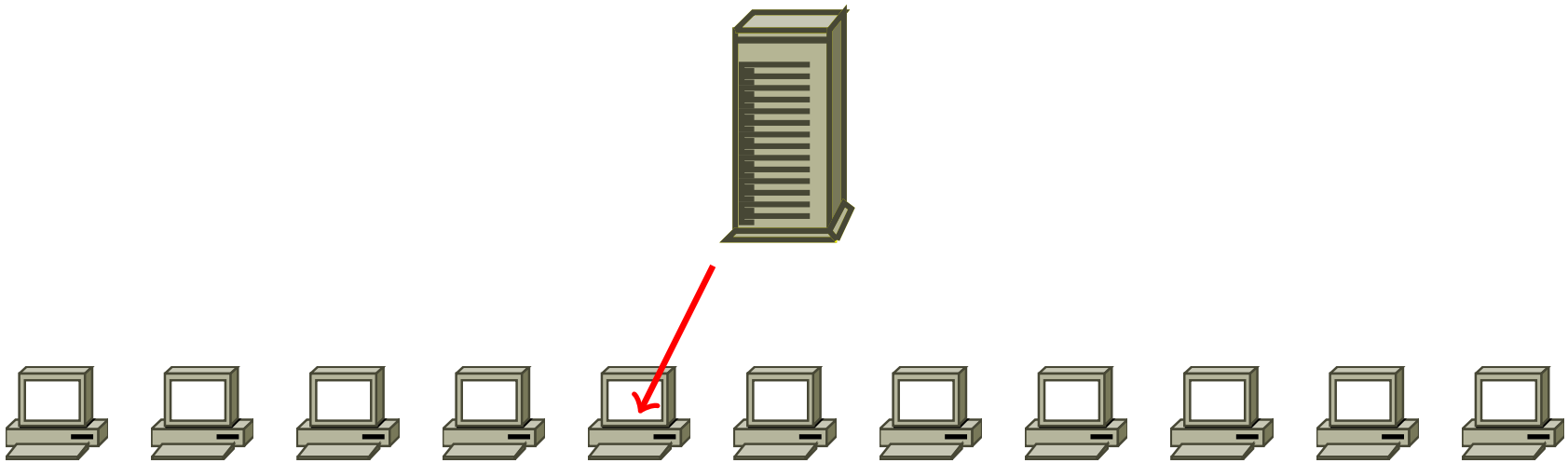
Sequential?

Scalable remote command execution with Taktuk



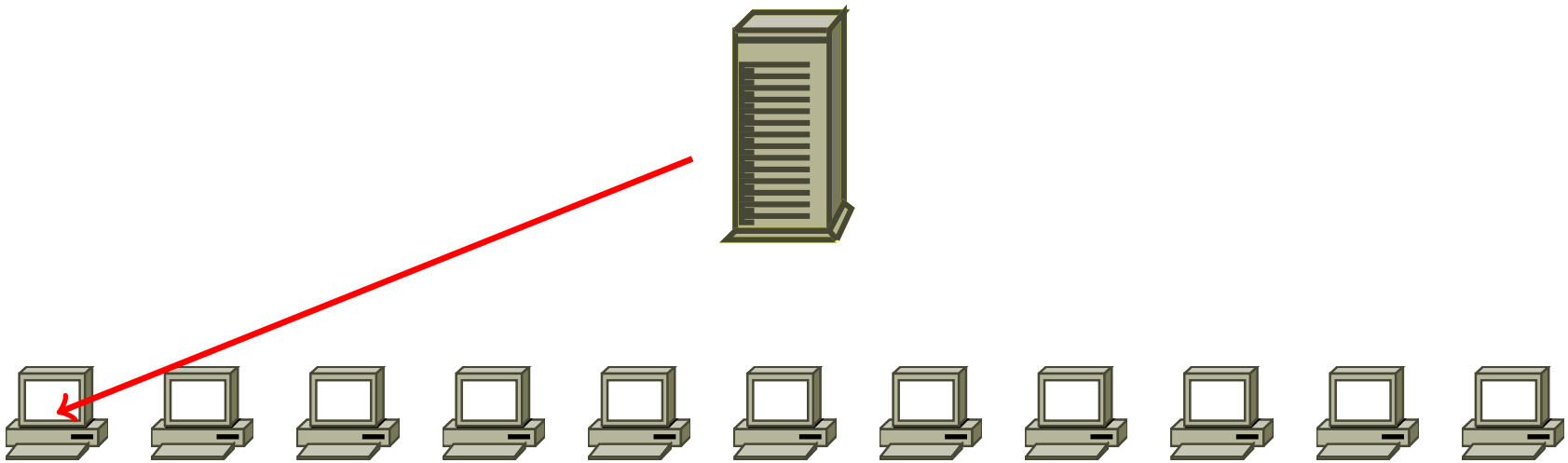
Sequential?

Scalable remote command execution with Taktuk



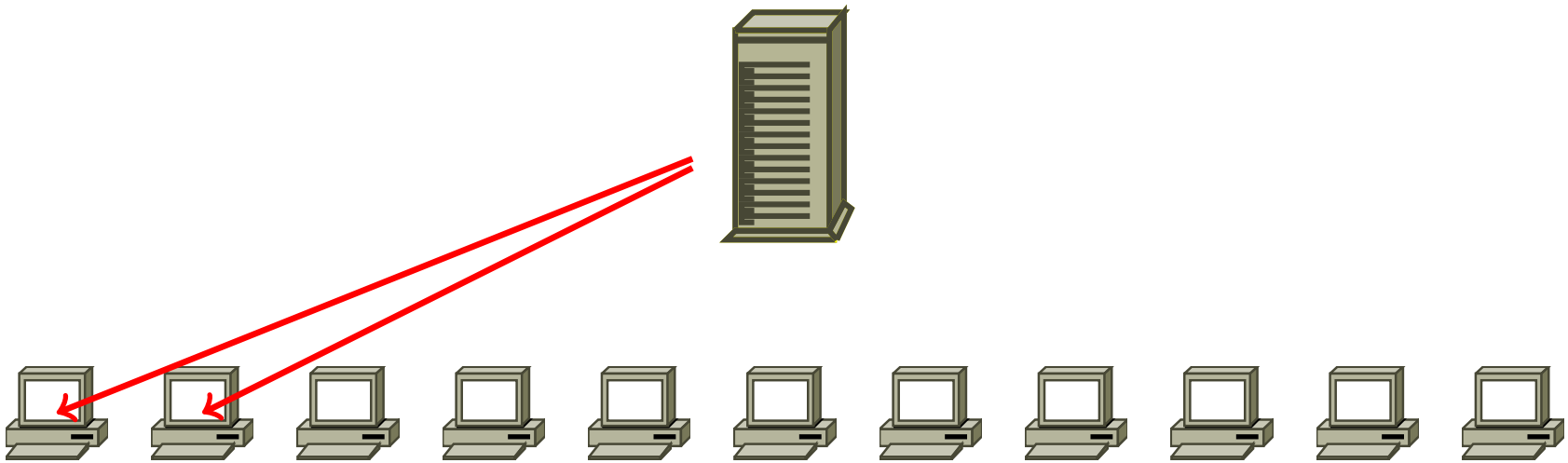
Sequential?

Scalable remote command execution with Taktuk



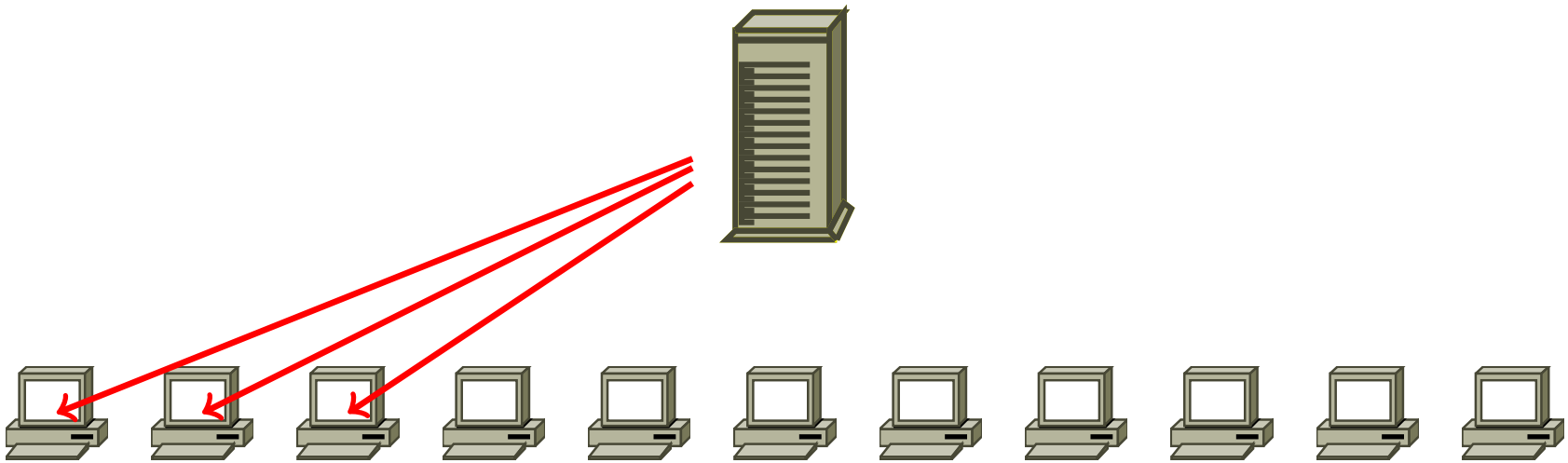
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



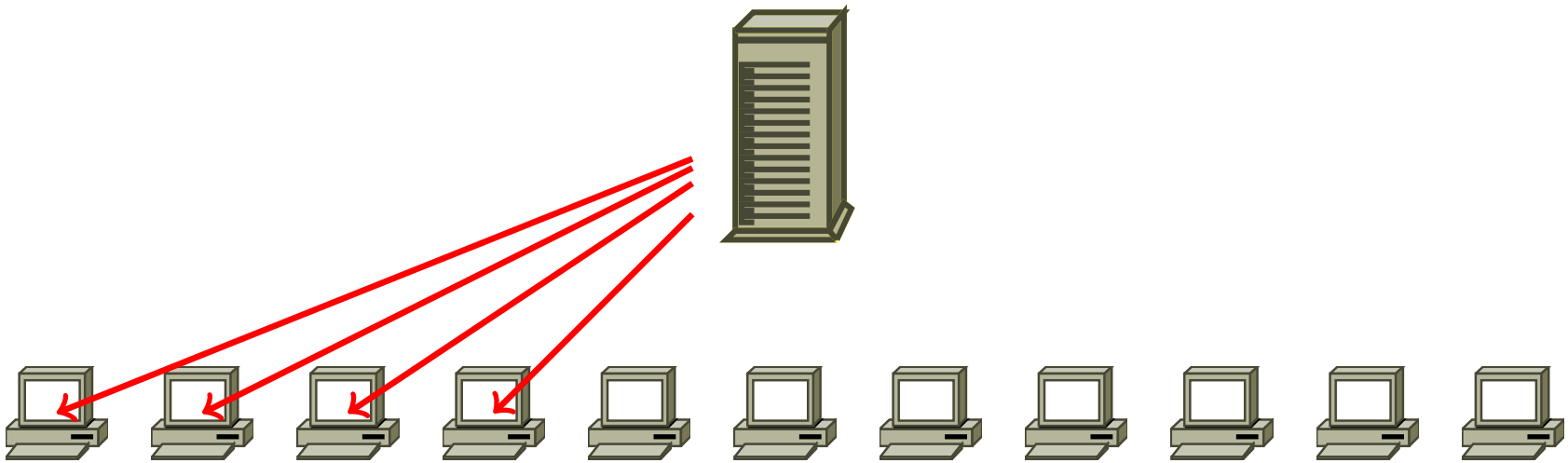
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



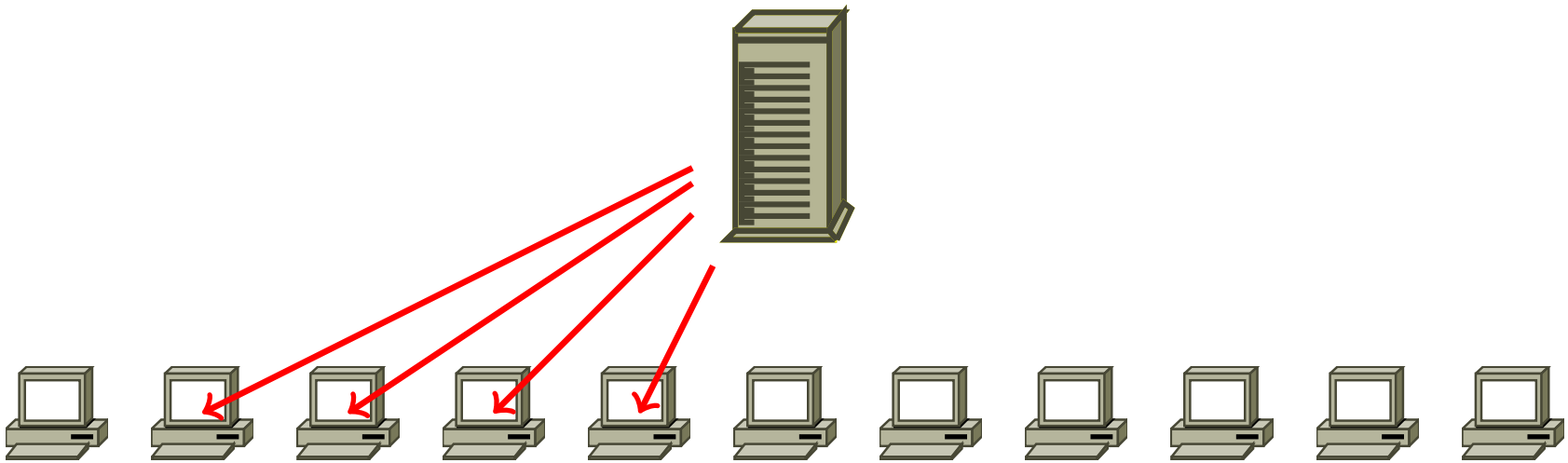
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



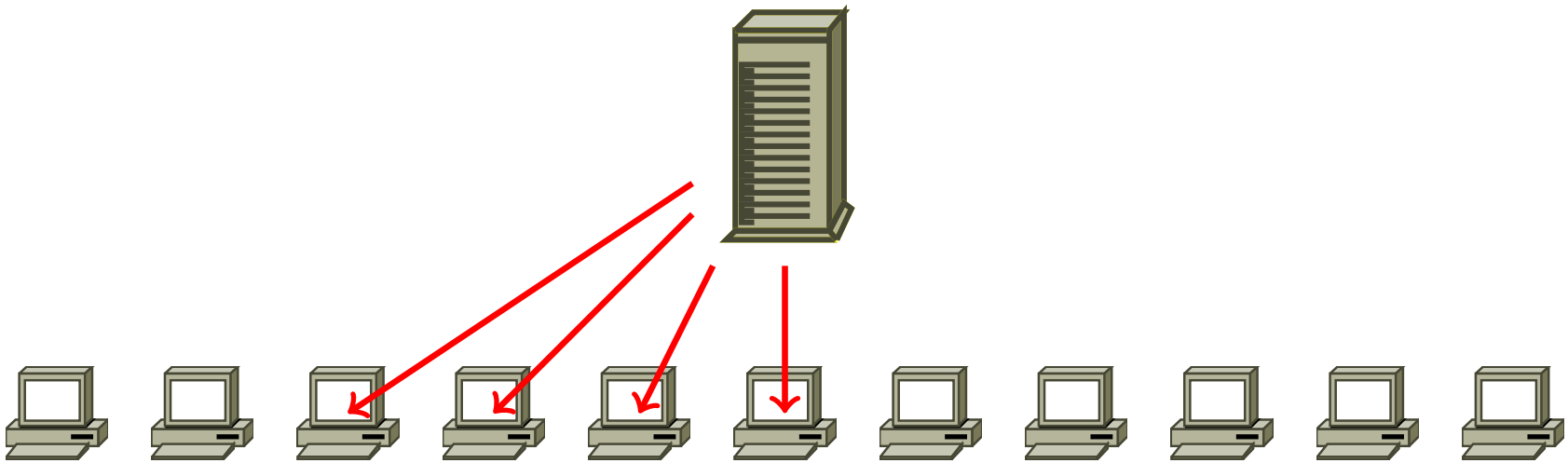
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



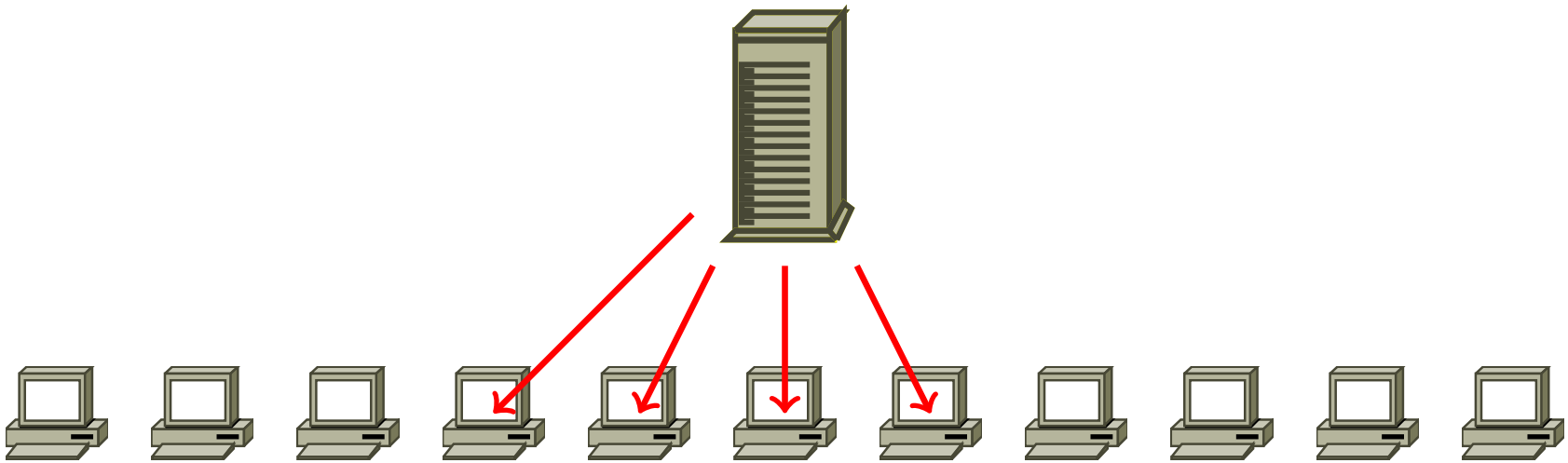
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



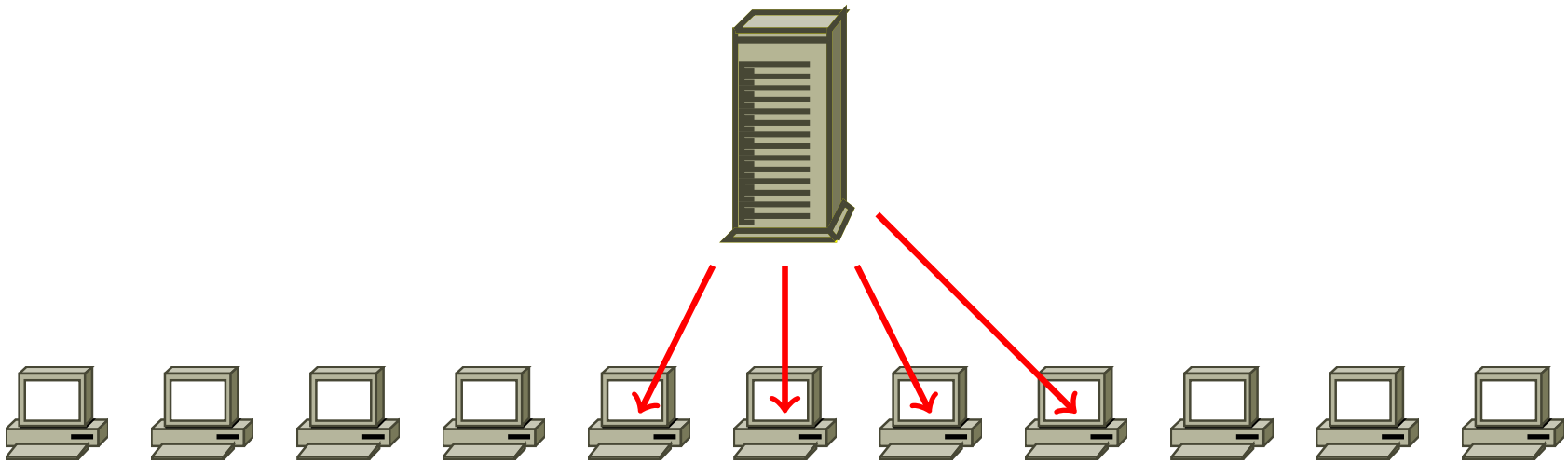
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



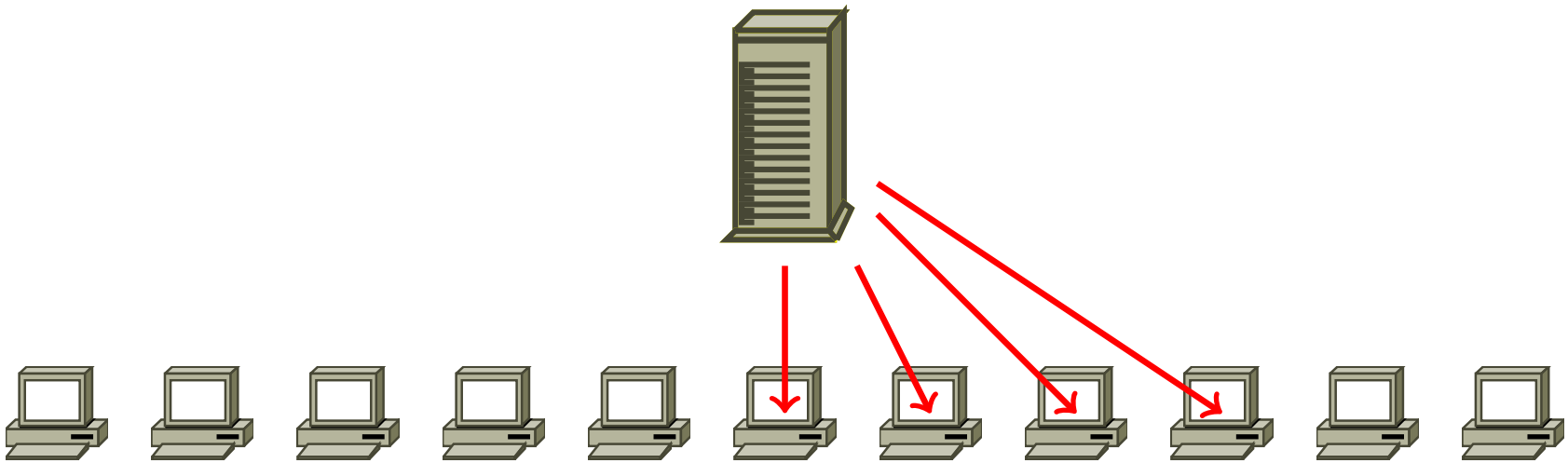
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



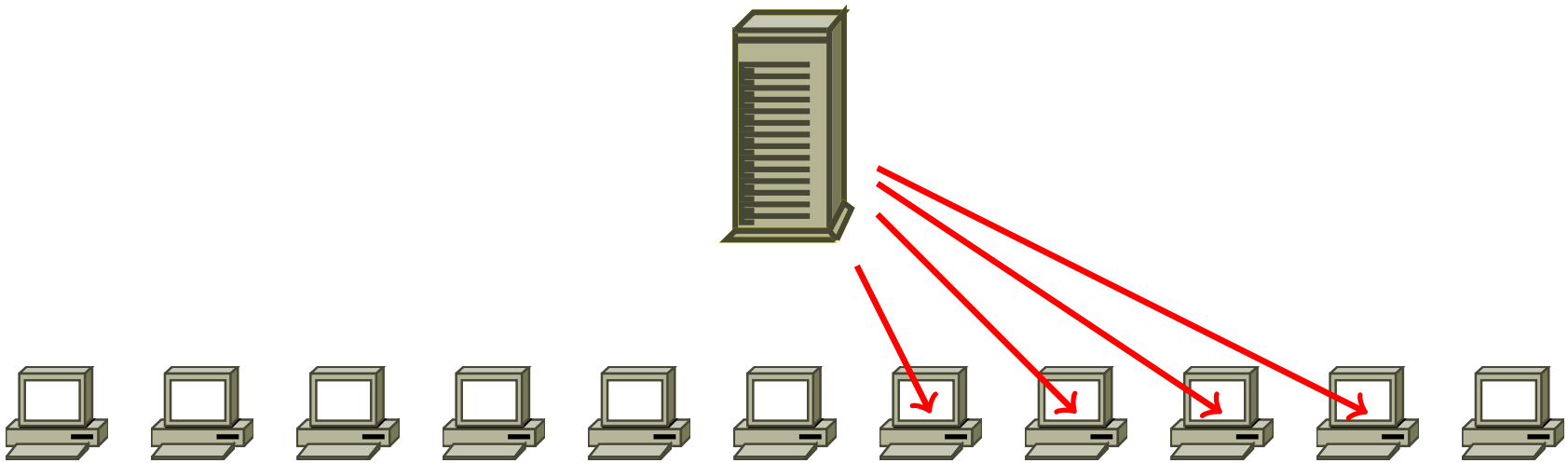
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



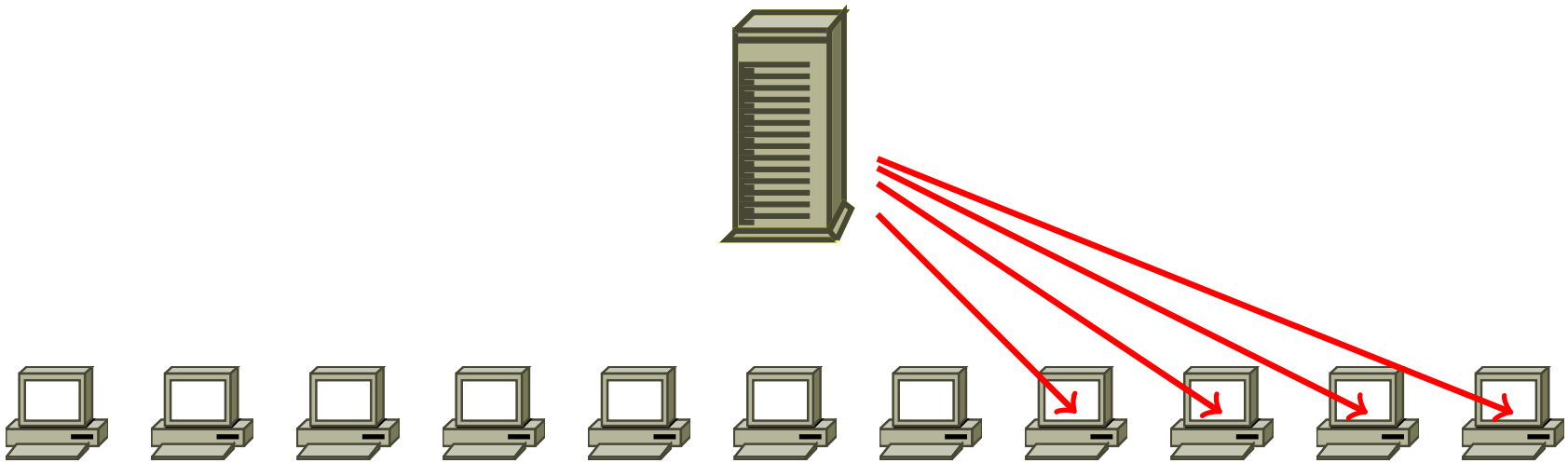
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



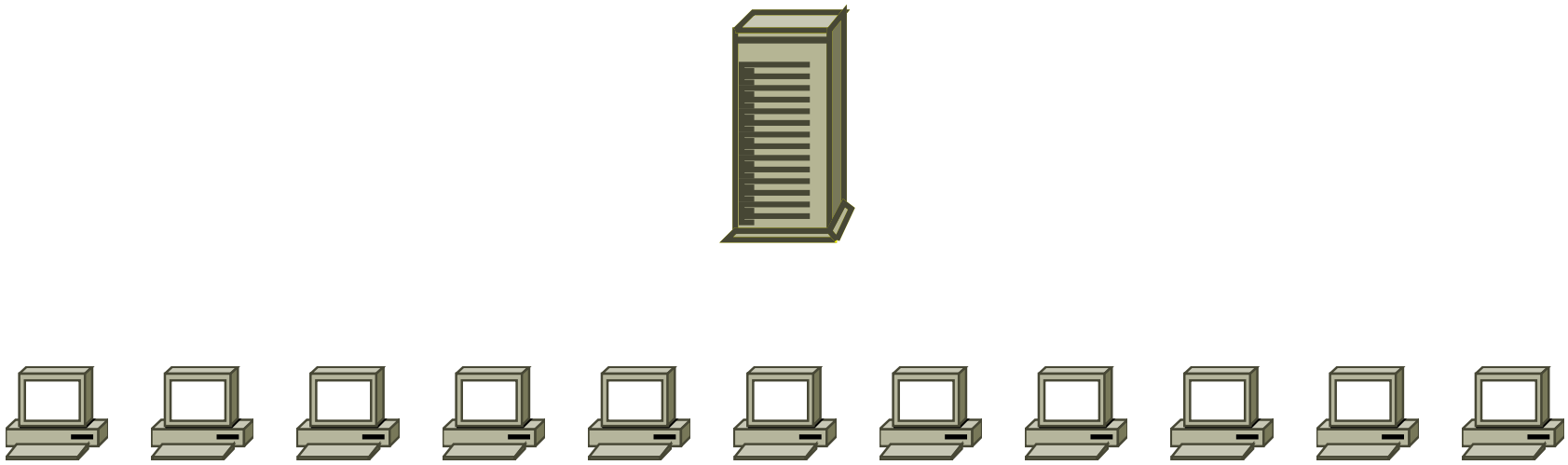
Sequential + sliding window (pdsh-like)?

Scalable remote command execution with Taktuk



Sequential + sliding window (pdsh-like)?

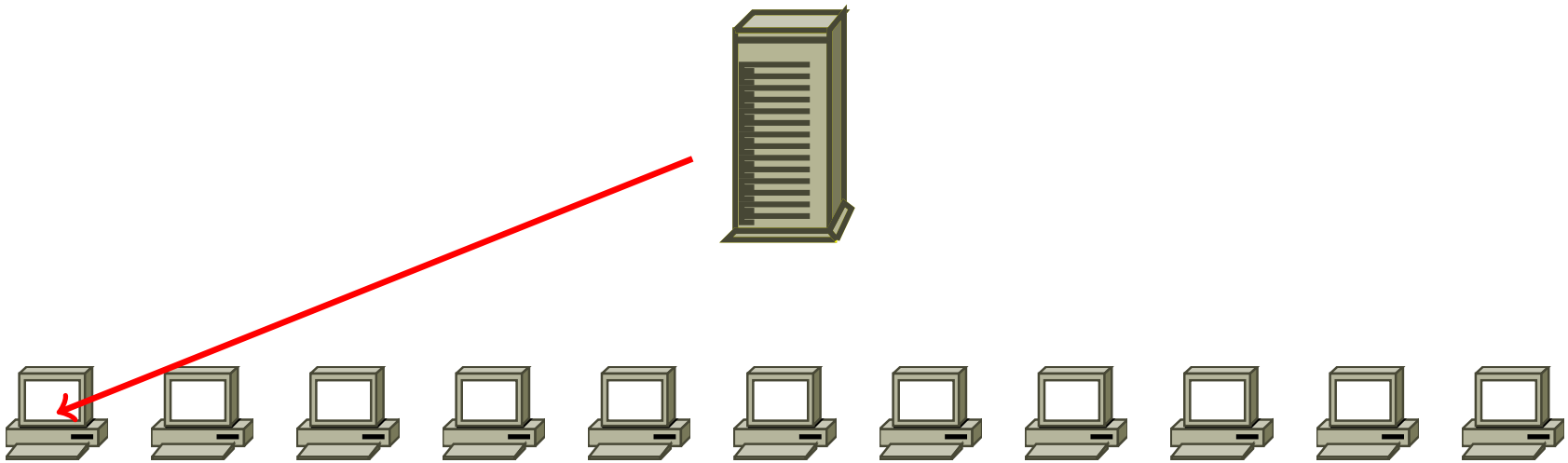
Scalable remote command execution with Taktuk



In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- using TakTuk – **http://taktuk.gforge.inria.fr/**
- HPDC'2009 paper:
  B. Claudel, G. Huard and O. Richard.
  TakTuk, Adaptive Deployment of Remote Executions.

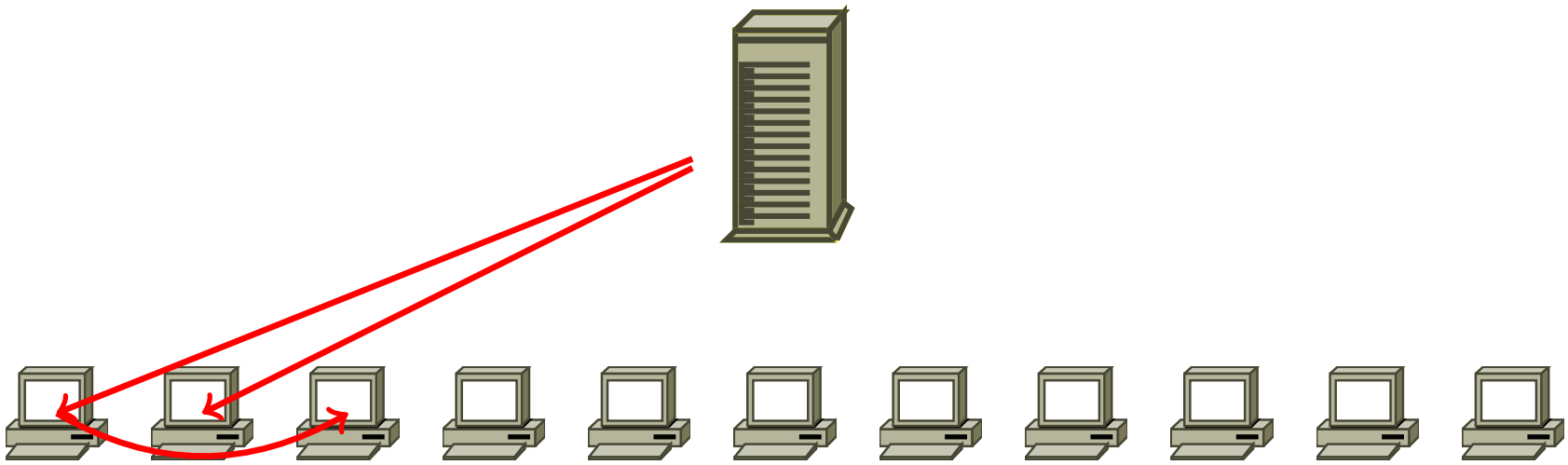
Scalable remote command execution with Taktuk



In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- using TakTuk – **http://taktuk.gforge.inria.fr/**
- HPDC'2009 paper:
  B. Claudel, G. Huard and O. Richard.
  TakTuk, Adaptive Deployment of Remote Executions.

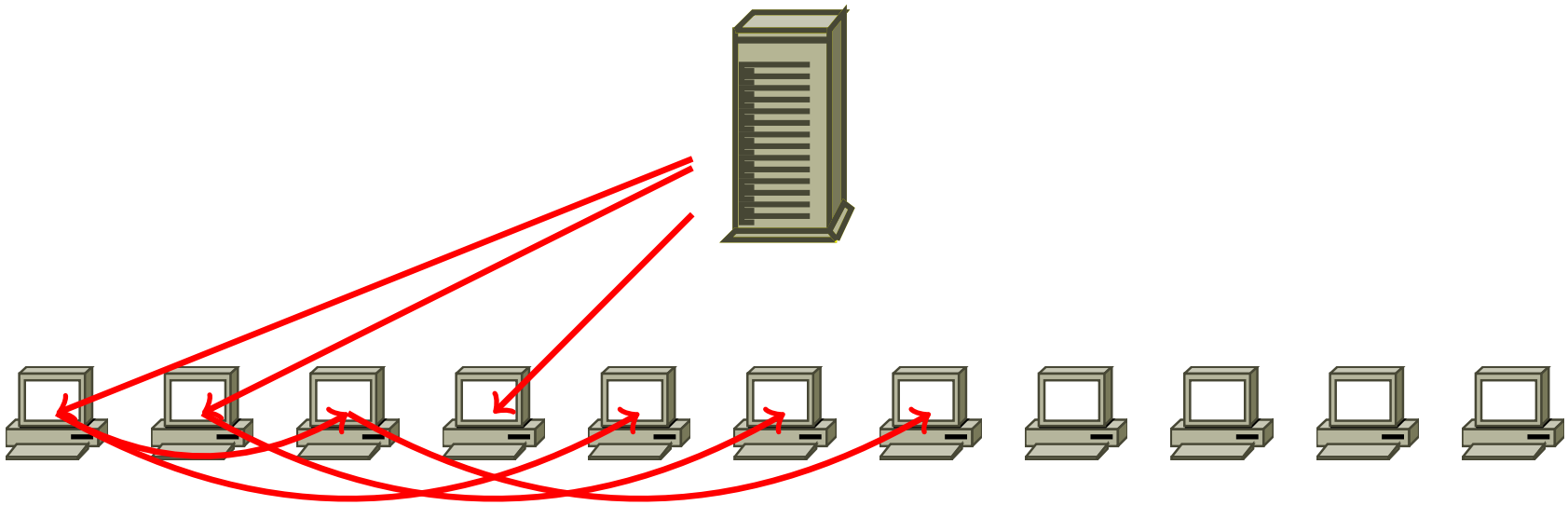
Scalable remote command execution with Taktuk



In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- using TakTuk – **http://taktuk.gforge.inria.fr/**
- HPDC'2009 paper:
  B. Claudel, G. Huard and O. Richard.
  TakTuk, Adaptive Deployment of Remote Executions.

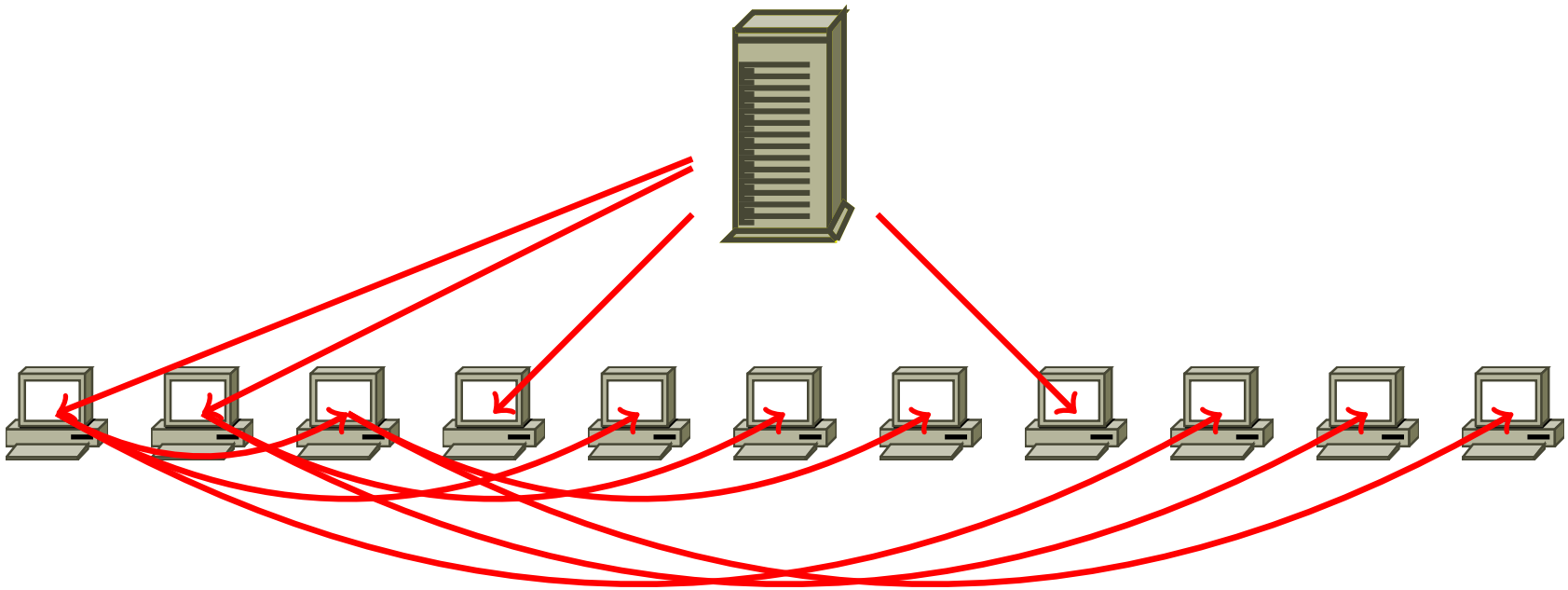
Scalable remote command execution with Taktuk



In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- using TakTuk – **http://taktuk.gforge.inria.fr/**
- HPDC'2009 paper:
  B. Claudel, G. Huard and O. Richard.
  TakTuk, Adaptive Deployment of Remote Executions.

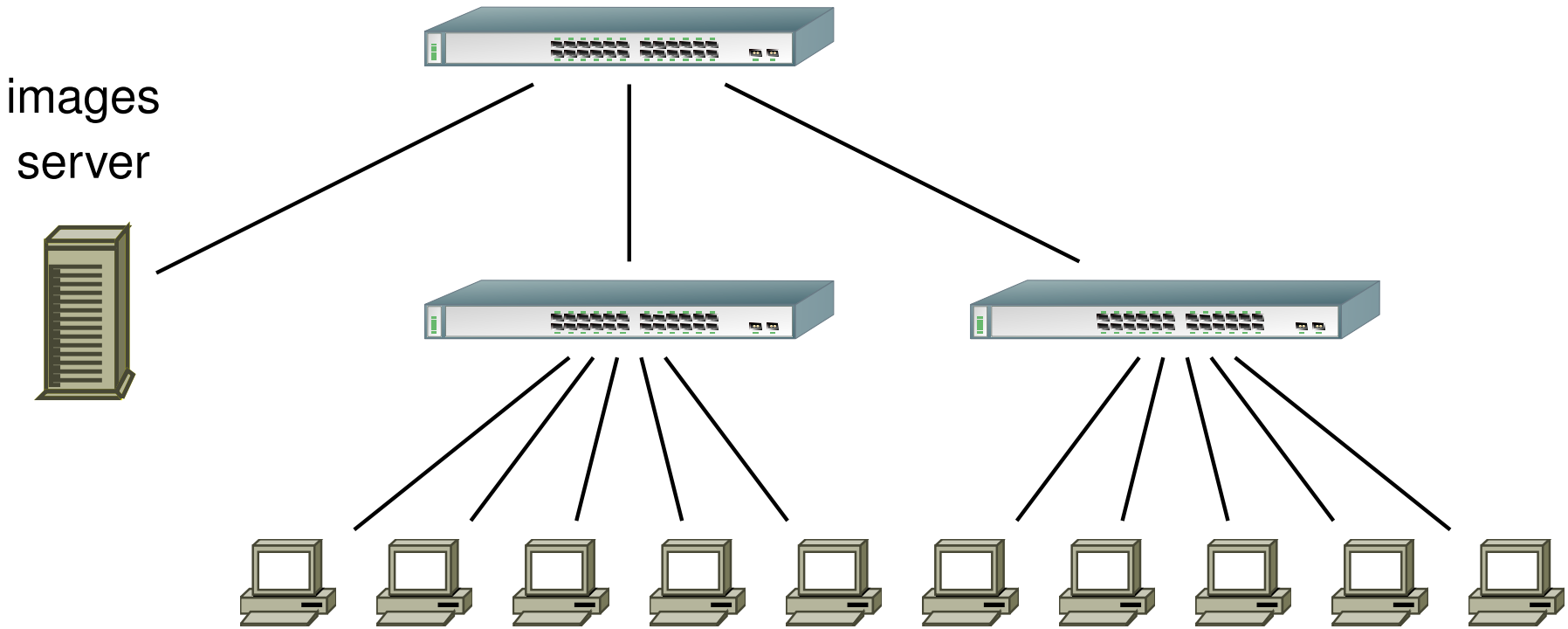
Scalable remote command execution with Taktuk



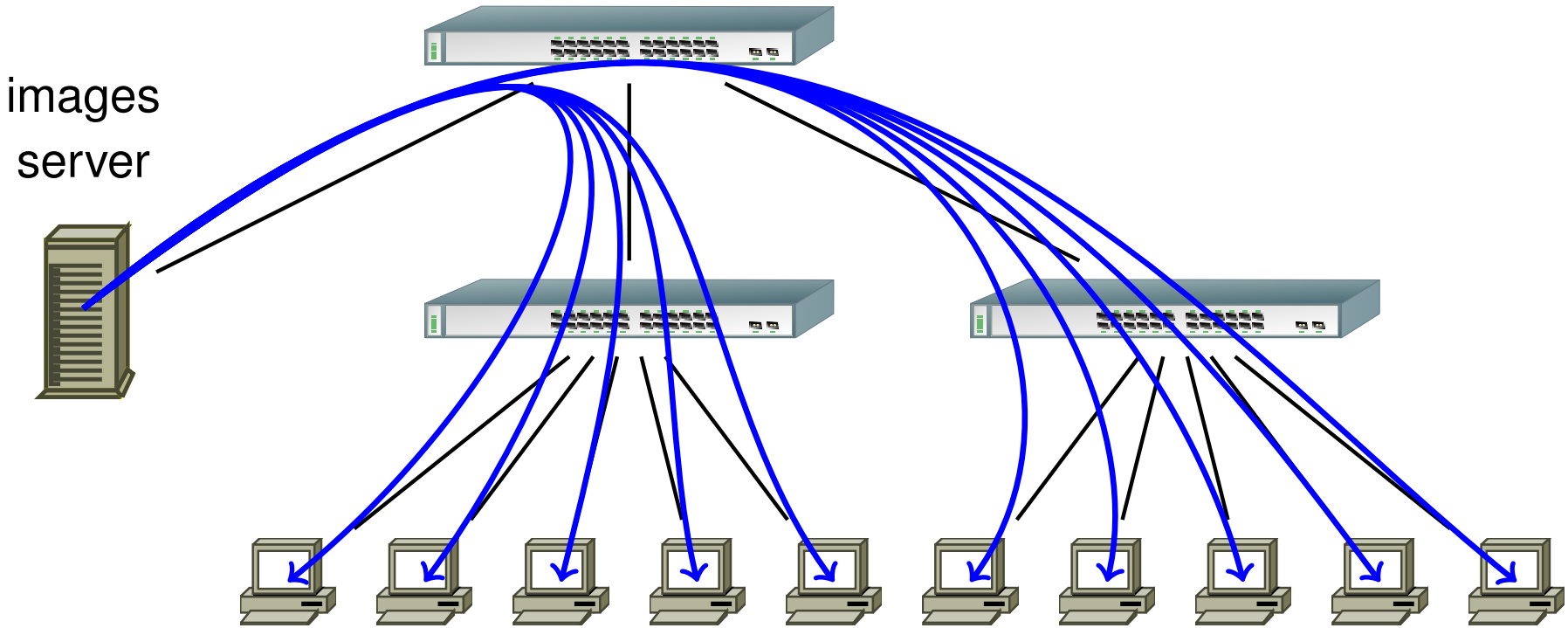
In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- using TakTuk – **http://taktuk.gforge.inria.fr/**
- HPDC'2009 paper:
  B. Claudel, G. Huard and O. Richard.
  TakTuk, Adaptive Deployment of Remote Executions.

Broadcast of system images

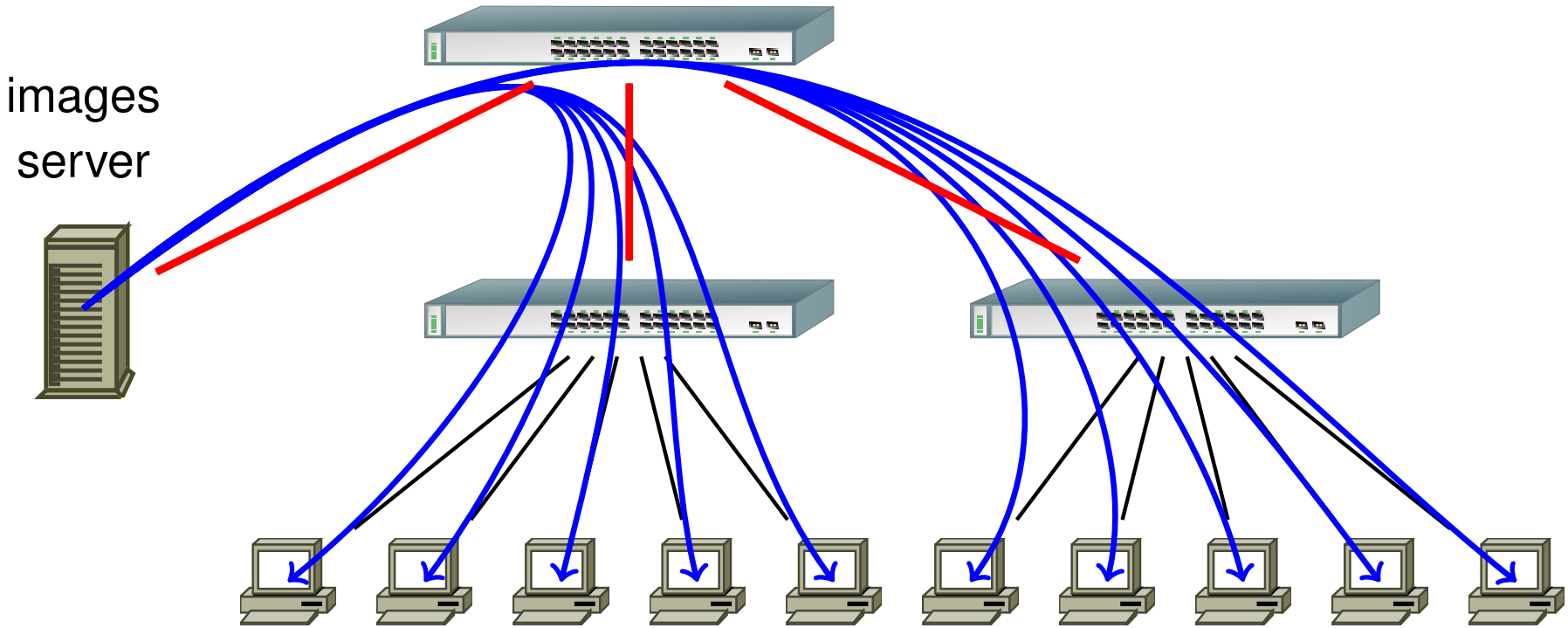


Broadcast of system images



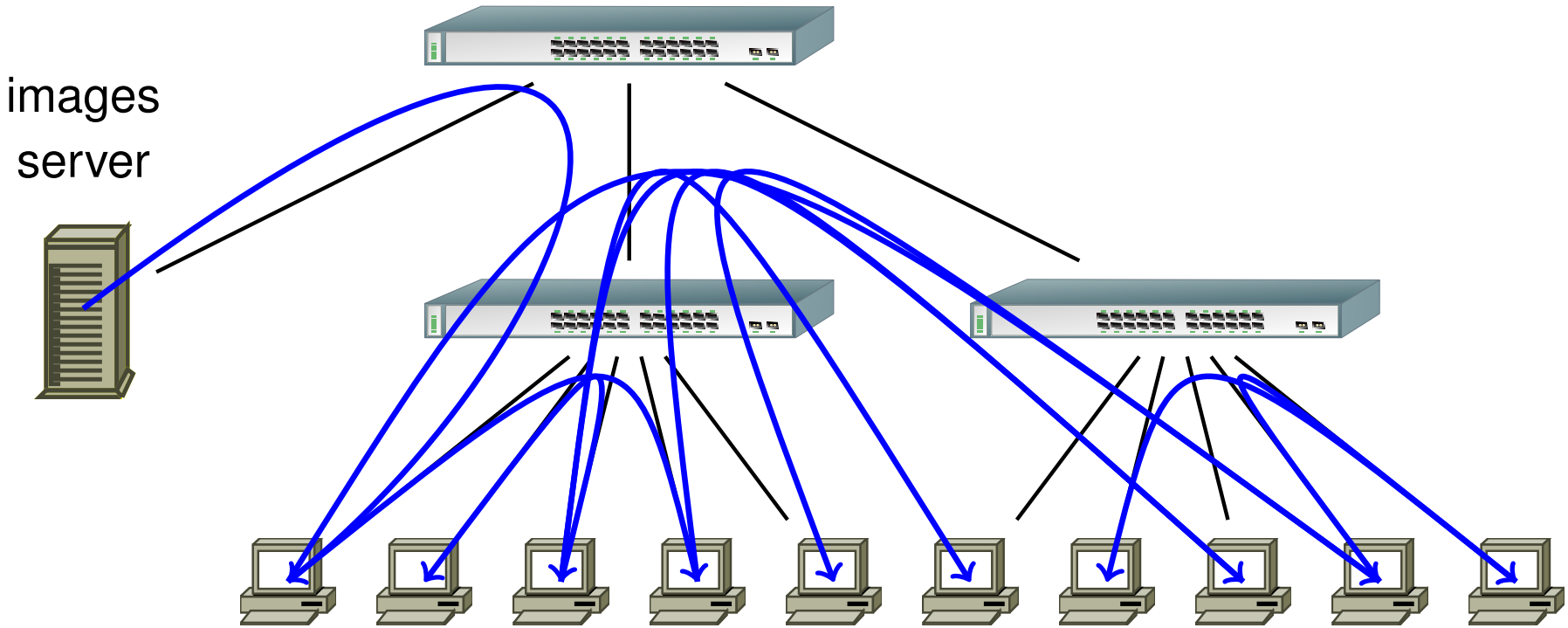
Send from server node to every client?

Broadcast of system images



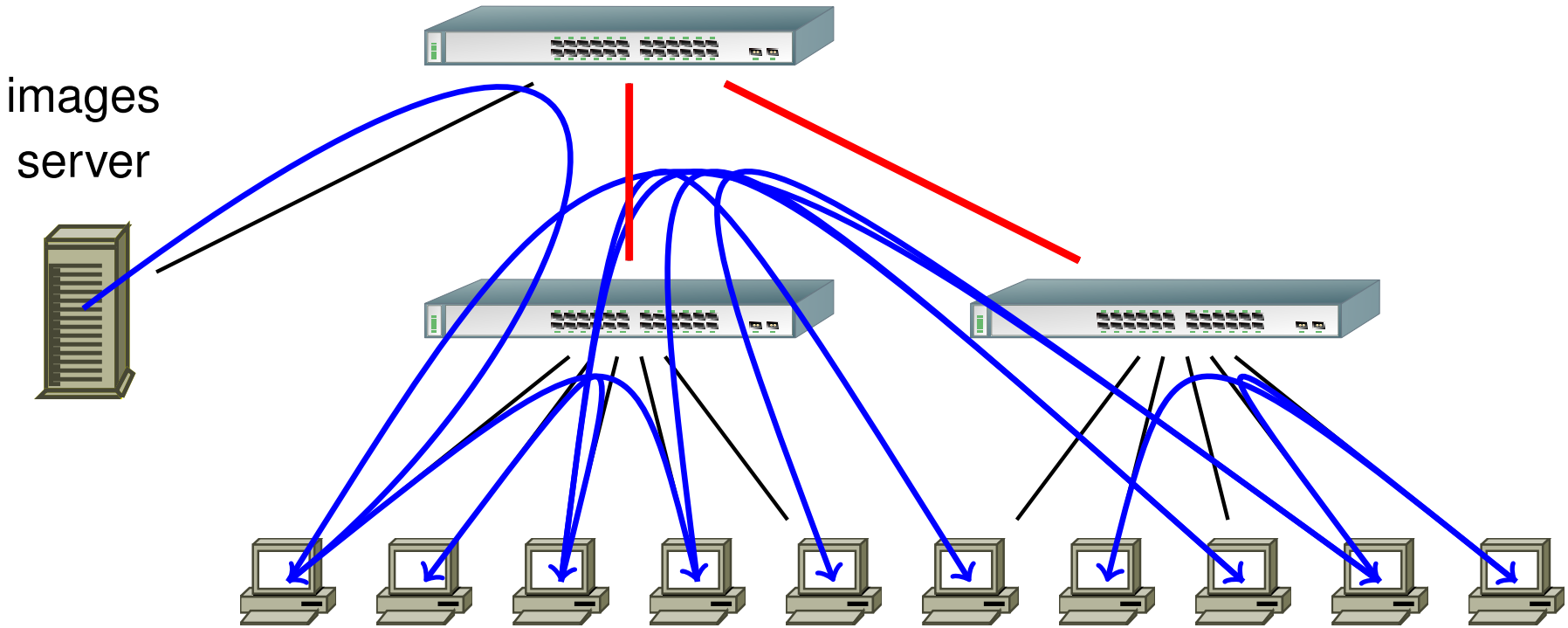
Send from server node to every client?

Broadcast of system images



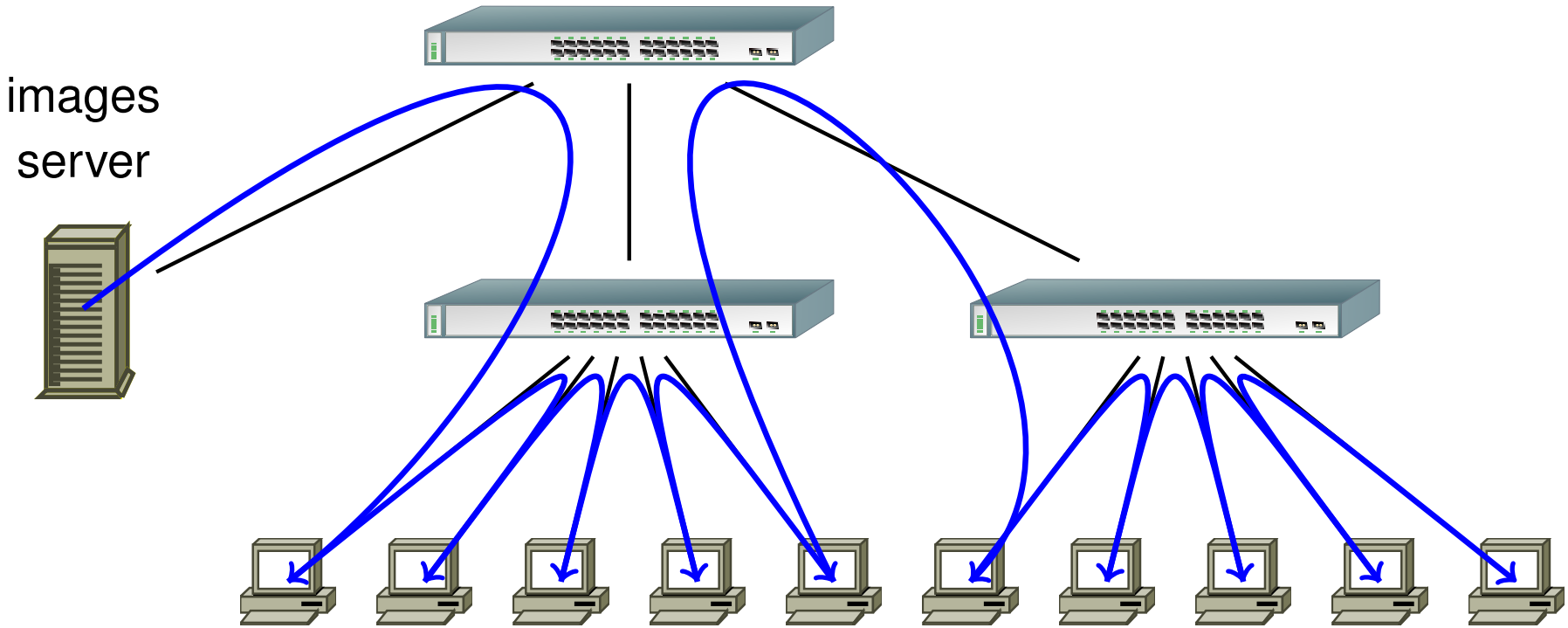
Use P2P?

Broadcast of system images



Use P2P?

Broadcast of system images



In Kadeploy: **Topology-aware chained broadcast**

- Limiting factor: backplane bandwidth of switches

Testing the scalability of Kadeploy

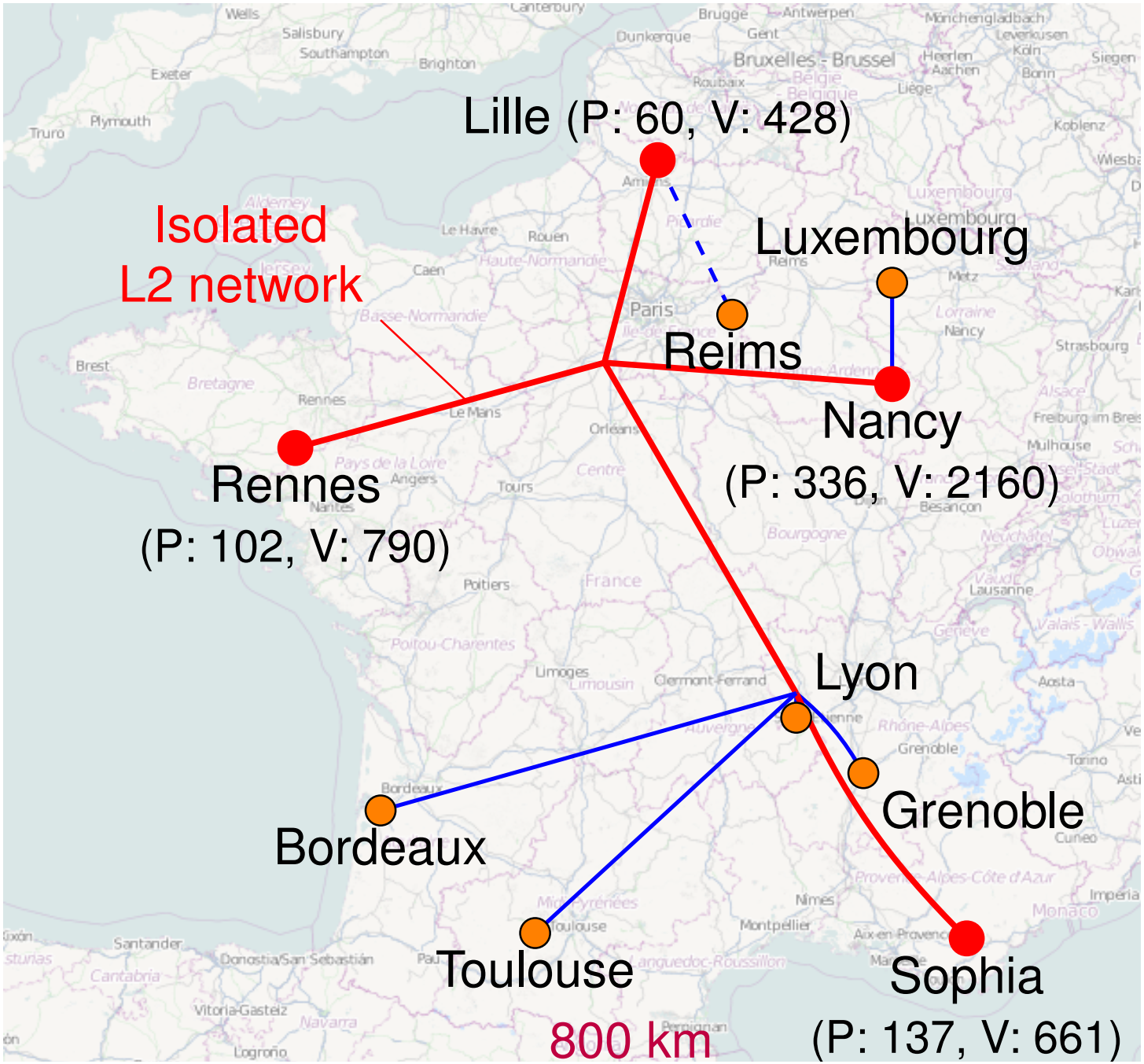
- **Rather specific requirements**
  - Many reinstallable nodes (infrastructure + deployed nodes)
  - DHCP server

Testing the scalability of Kadeploy

- **Rather specific requirements**
  - Many reinstallable nodes (infrastructure + deployed nodes)
  - DHCP server
- **Testbed: Grid'5000 - http://www.grid5000.fr/**
  - Testbed for research on distributed systems: HPC, Grids, P2P, Cloud
  - 10 sites, 25 clusters, 1300 nodes, 7400 cores
  - Unique features including:
    - *Hardware-as-a-Service Cloud*: redeployment of OS on the bare metal by users (using Kadeploy)
    - Dedicated backbone network
    - KaVLAN: network isolation

Testing the scalability of Kadeploy

- **Rather specific requirements**
  - Many reinstallable nodes (infrastructure + deployed nodes)
  - DHCP server
- **Testbed: Grid'5000 - http://www.grid5000.fr/**
  - Testbed for research on distributed systems: HPC, Grids, P2P, Cloud
  - 10 sites, 25 clusters, 1300 nodes, 7400 cores
  - Unique features including:
    - *Hardware-as-a-Service Cloud*: redeployment of OS on the bare metal by users (using Kadeploy)
    - Dedicated backbone network
    - KaVLAN: network isolation
- **Still not enough nodes $\rightsquigarrow$ virtual machines (KVM) on all nodes**



3-18 VM
per node

Totals:

Physical: 635

Virtual: 3999

Experimental process (fully automated)

1. Virtual testbed preparation
   - Reserve and reinstall all nodes $\rightsquigarrow$ 20 mins
   - Prepare 33 infrastructure nodes and 635 VM-hosting nodes; configure everything; start virtual machines $\rightsquigarrow$ 20 mins
2. One or more Kadeploy runs
   - e.g. 3999 virtual nodes (3838 successful) $\rightsquigarrow$ 57 mins
   - Hotspots:
     - First reboot: 11 mins
     - Broadcast: 15 mins
     - Second reboot: 7 mins

Limits to scalability and future work

Two major limiting factors:

- Nodes reboot
  - Relies on unreliable protocols: DHCP, TFTP (HTTP if iPXE)
  - Mitigated in Kadeploy by using reboot windows
- Remote command execution and broadcast of system image
  - Heavily stresses the network $\rightsquigarrow$ ARP and TCP timeouts
    - Dynamic TakTuk tree $\rightsquigarrow$ more ARP needed
    - Large Cloud infrastructures use per-rack L2 networks
  - Future work:
    - Robustify ARP and TCP (iPXE+kernel tuning)
    - Improve fault tolerance of image broadcast
    - Infiniband support

Conclusions

- Tested the scalability of the Kadeploy OS provisioning solution
  - Critical service in cluster environments
- Configured a *Cloud* of KVM virtual machines:
  - Using our own VM management scripts
  - Of 3999 virtual machines
  - On 668 physical machines
  - From 4 sites of the Grid'5000 testbed
  - In a L2 network spanning 1000 km
- Reinstalled those virtual machines using Kadeploy
  $<$ 1 hour for 3838 machines successfully installed
- Fully automated process; no special Grid'5000 privileges required
- Identified several bottlenecks and ideas for future work