Kadeploy

From Scalable and Reliable Bare-metal Provisioning to a Reconfigurable Experimental Testbed

Lucas Nussbaum

lucas.nussbaum@loria.fr

Joint work with Luc Sarzyniec and Emmanuel Jeanvoine

Kadeploy: an OS provisioning solution

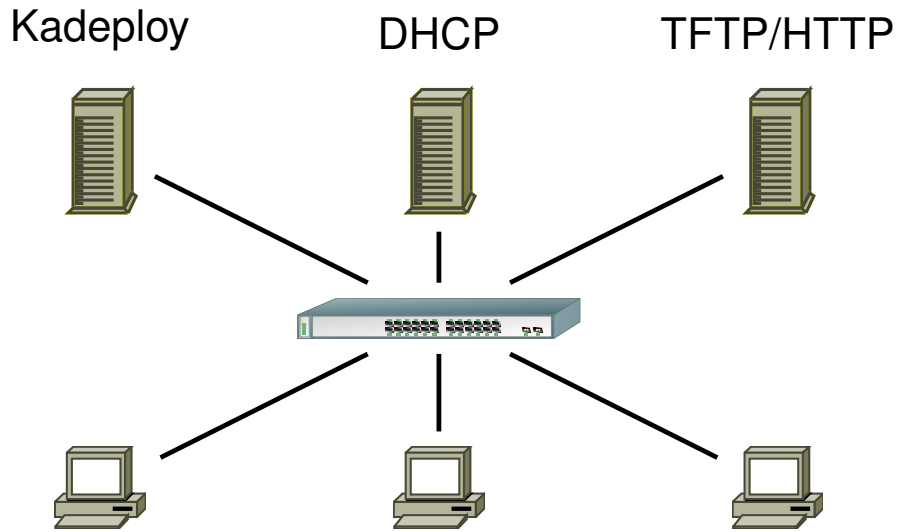
- Install compute nodes using system images
  - Similar problem space to CloneZilla, SystemImager, xCAT, Ironic
- Designed for scalability and reliability
- **Debian** and **RPM** packages, active development since 2004
- CeCILL v2 license (GPL&AGPL compatible – see 5.3.4)

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

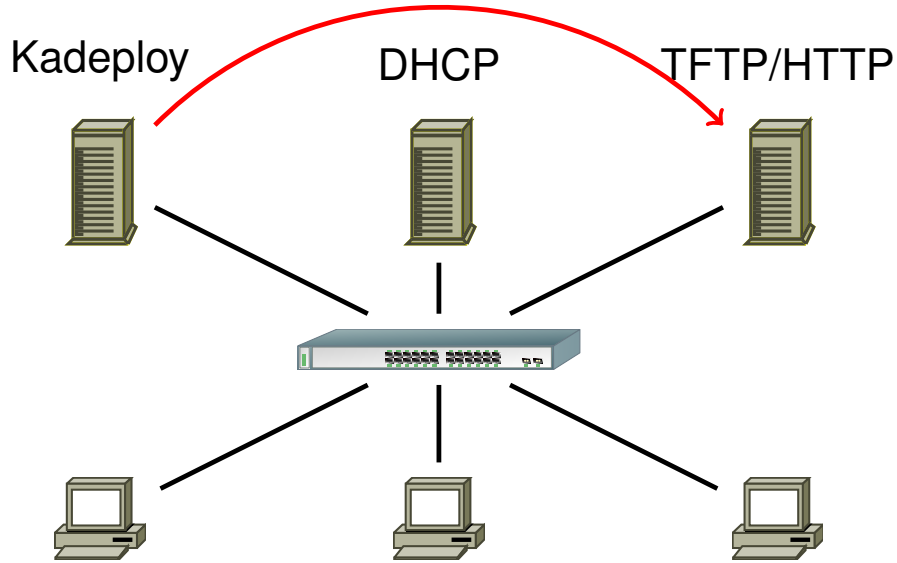
Key features

- **Install and configure a large number of nodes**
  - Install several cluster in one shot, from a single client
  - Support for concurrent deployments
- **Manage a library of pre-configured system images**
  - User-provided images, permissions management
- **Ecosystem**
  - Built on top of PXE, DHCP, TFTP/HTTP, SSH
  - Customizable remote low-level operations (IPMI, etc.)
  - Integration with batch scheduler and network isolation tools
- **Support for basically any operating system**
  (Linux, *BSD, Windows, ...)
- **Remote control API (REST)**
- **Fast**: 200 nodes $\rightsquigarrow$ 3 minutes

Kadeploy process overview

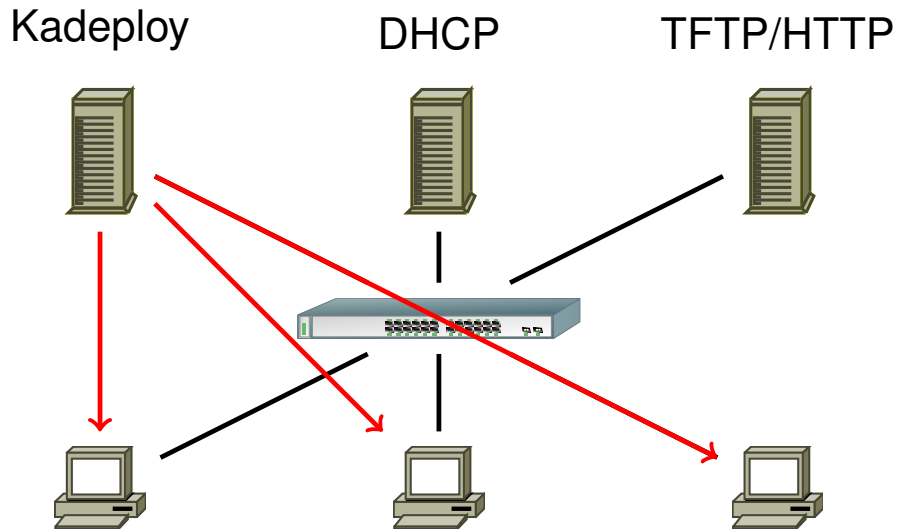


Kadeploy process overview



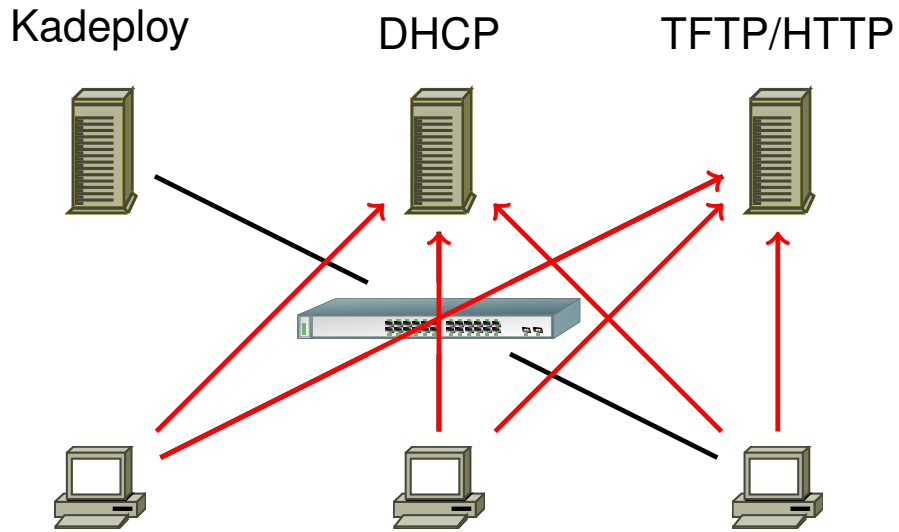
1. Kadeploy configures PXE profiles

Kadeploy process overview



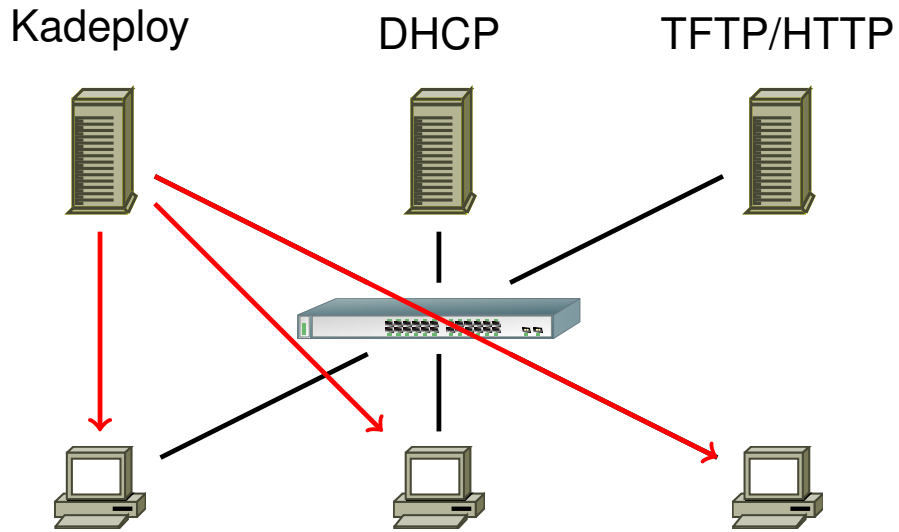
1. Kadeploy configures PXE profiles
2. Kadeploy triggers reboot using IPMI or SSH

Kadeploy process overview



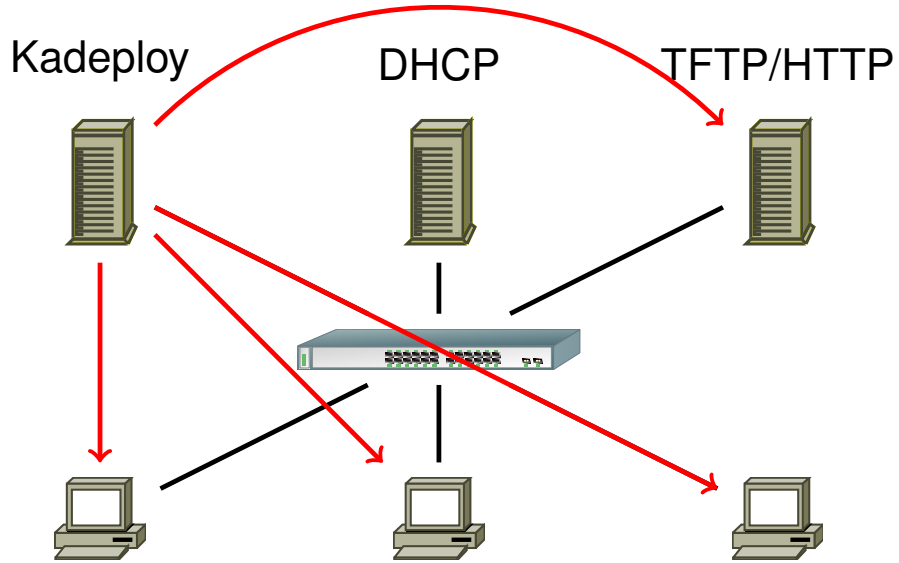
1. Kadeploy configures PXE profiles
2. Kadeploy triggers reboot using IPMI or SSH
3. Nodes boot to minimal deployment system sent over the network

Kadeploy process overview



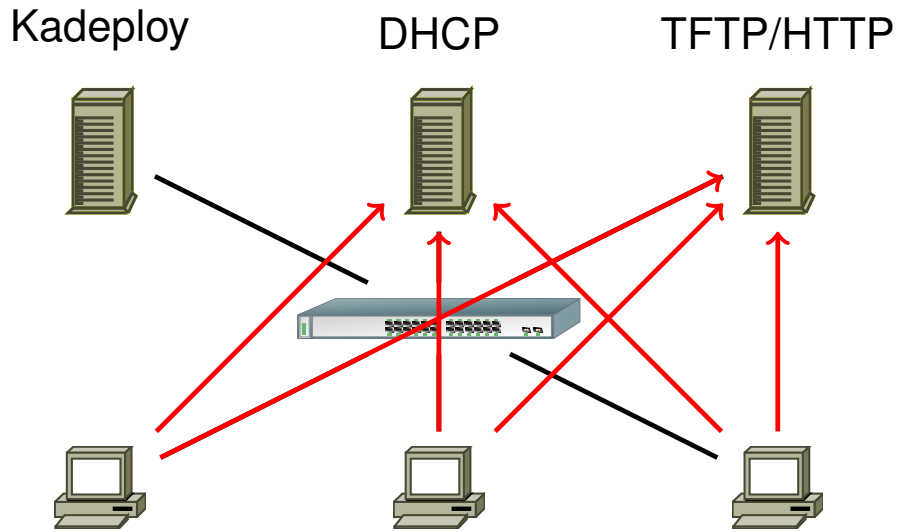
1. Kadeploy configures PXE profiles
2. Kadeploy triggers reboot using IPMI or SSH
3. Nodes boot to minimal deployment system sent over the network
4. Kadeploy configures nodes and sends system image

Kadeploy 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

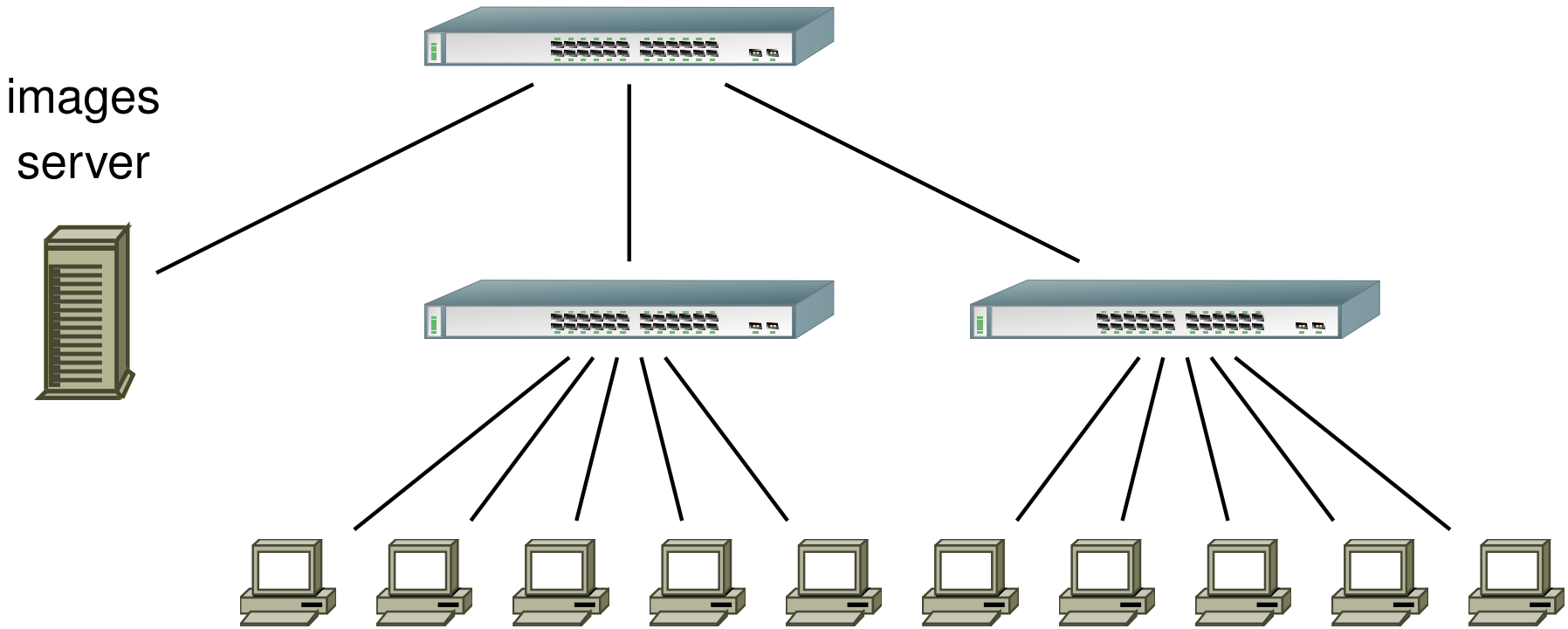
Kadeploy process overview



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

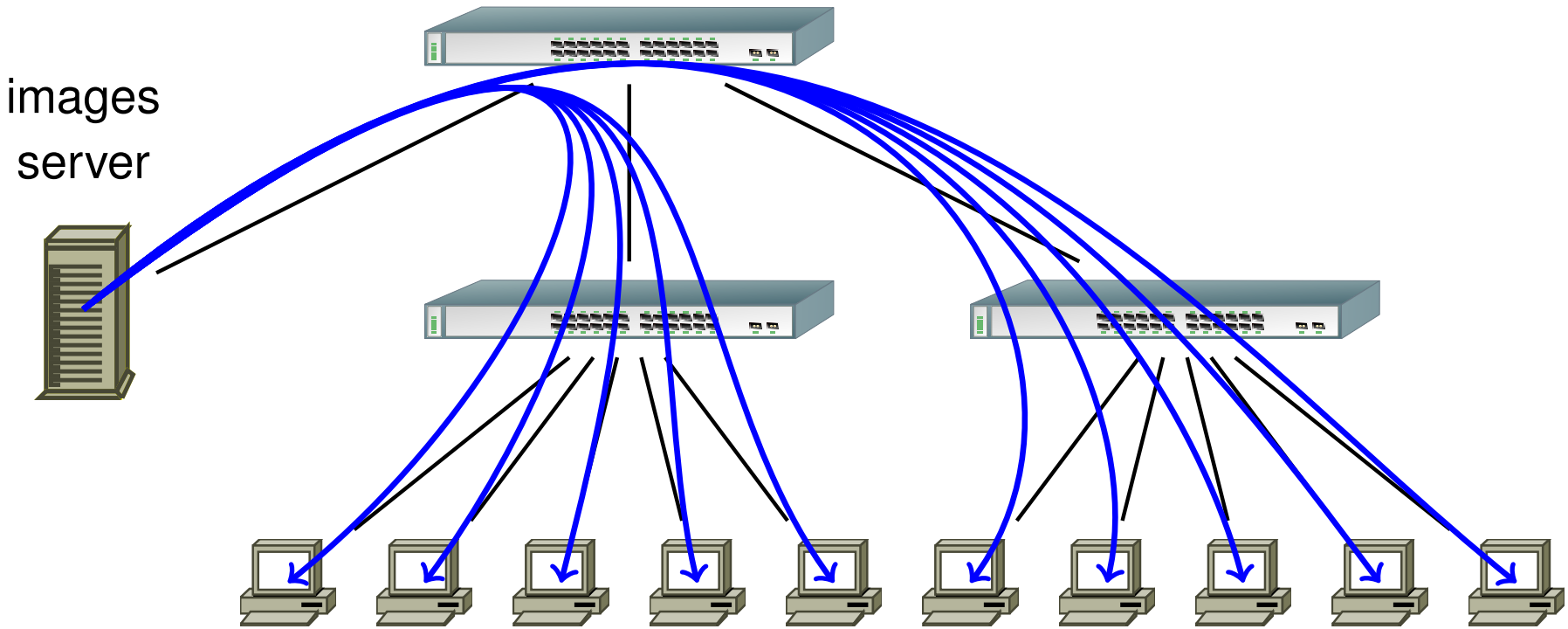
Scalable system image broadcast

- **Goal:** send a large amount of data to thousands of nodes
- **Challenge:** avoid network bottlenecks, saturation of links



Scalable system image broadcast

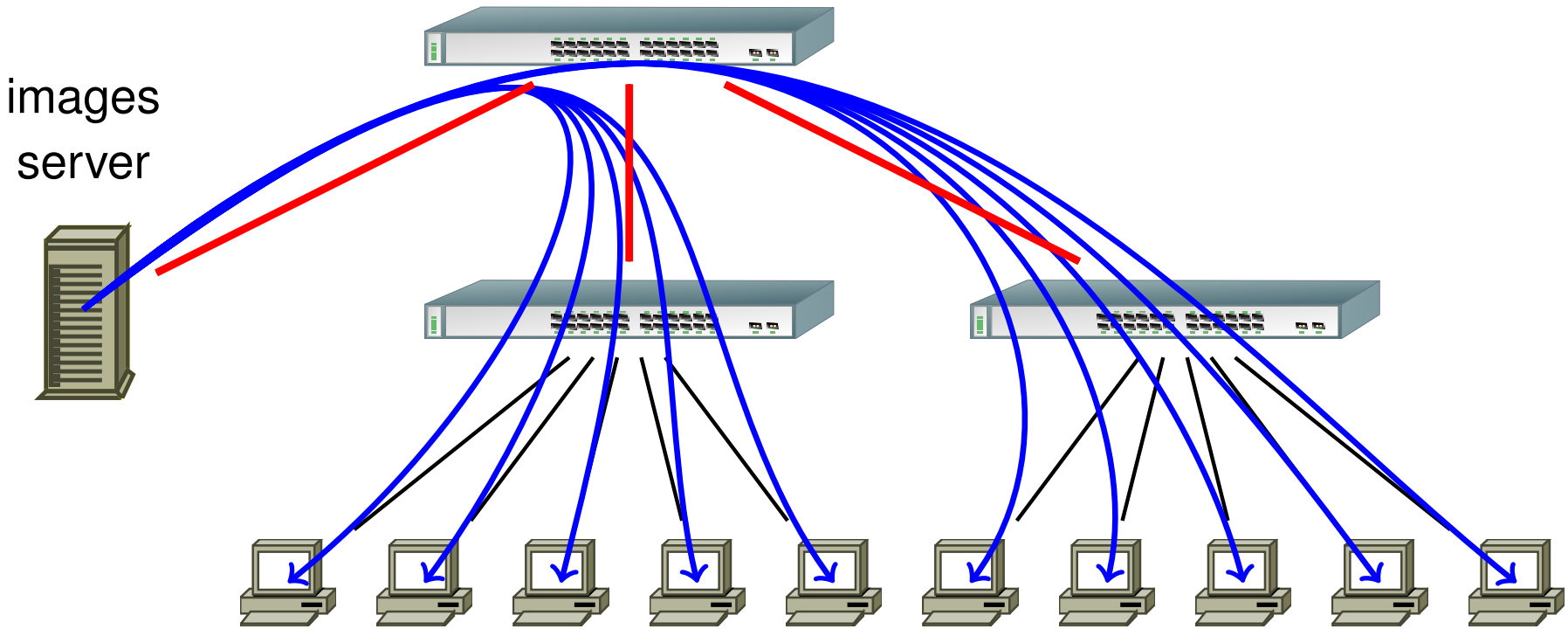
- **Goal:** send a large amount of data to thousands of nodes
- **Challenge:** avoid network bottlenecks, saturation of links



Send from server node to every client?

Scalable system image broadcast

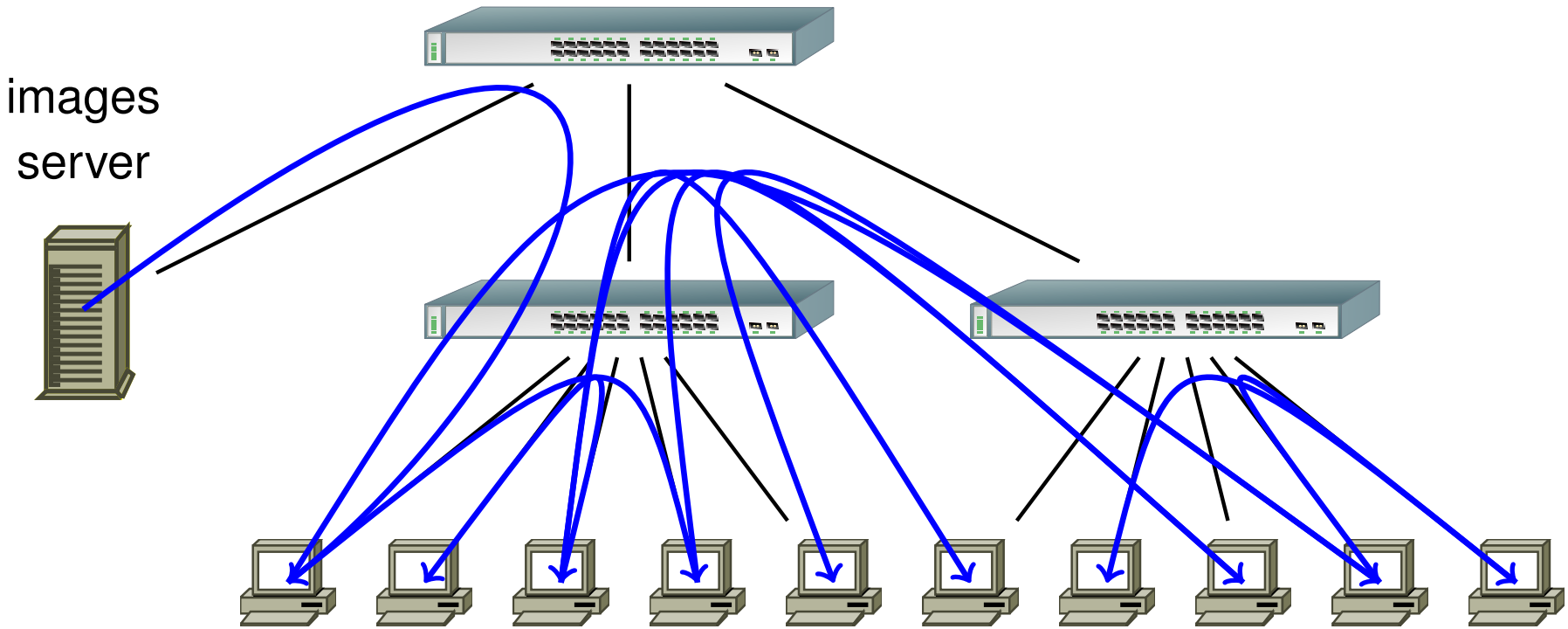
- **Goal:** send a large amount of data to thousands of nodes
- **Challenge:** avoid network bottlenecks, saturation of links



Send from server node to every client?

Scalable system image broadcast

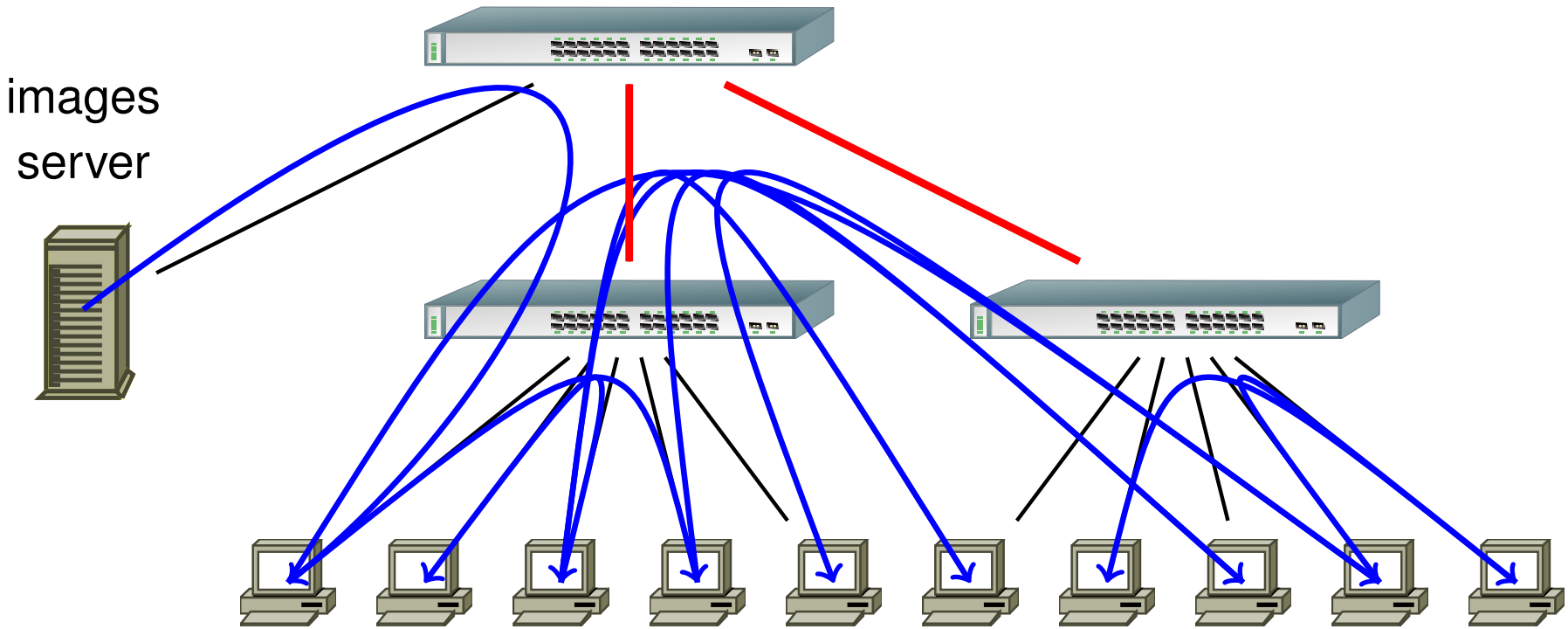
- **Goal:** send a large amount of data to thousands of nodes
- **Challenge:** avoid network bottlenecks, saturation of links



Use P2P?

Scalable system image broadcast

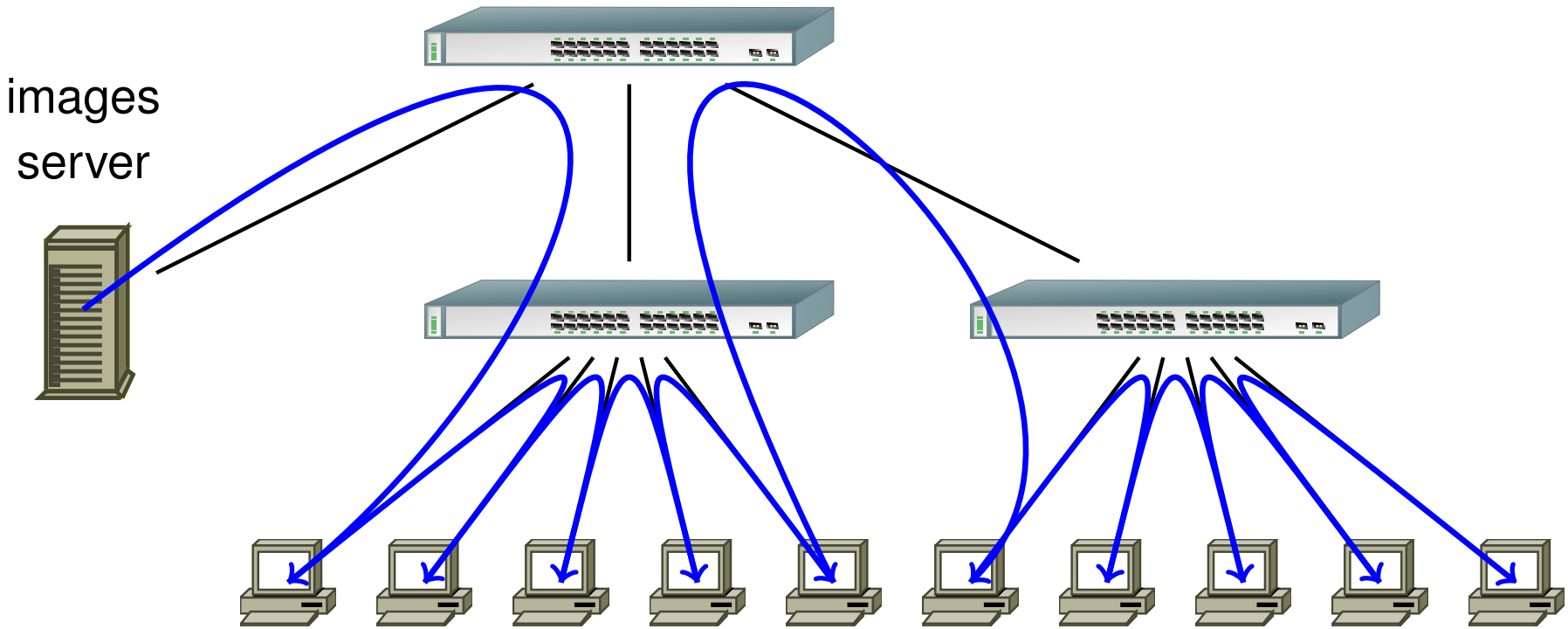
- **Goal:** send a large amount of data to thousands of nodes
- **Challenge:** avoid network bottlenecks, saturation of links



Use P2P?

Scalable system image broadcast

- **Goal:** send a large amount of data to thousands of nodes
- **Challenge:** avoid network bottlenecks, saturation of links

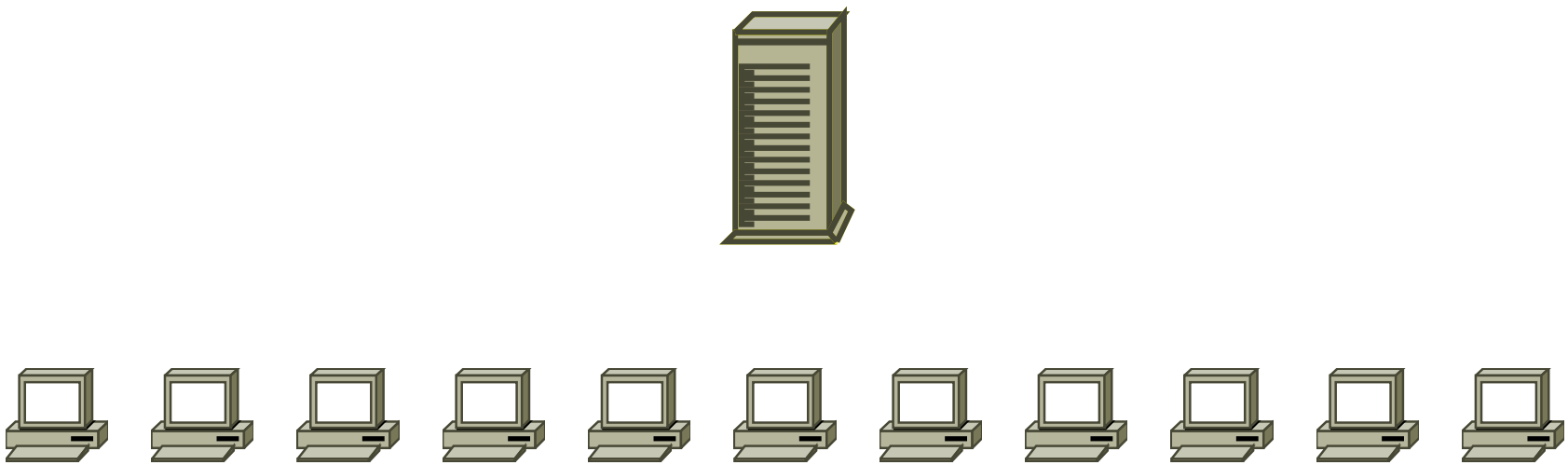


In Kadeploy: **Topology-aware pipelined broadcast**

- Limiting factor: backplane bandwidth of switches

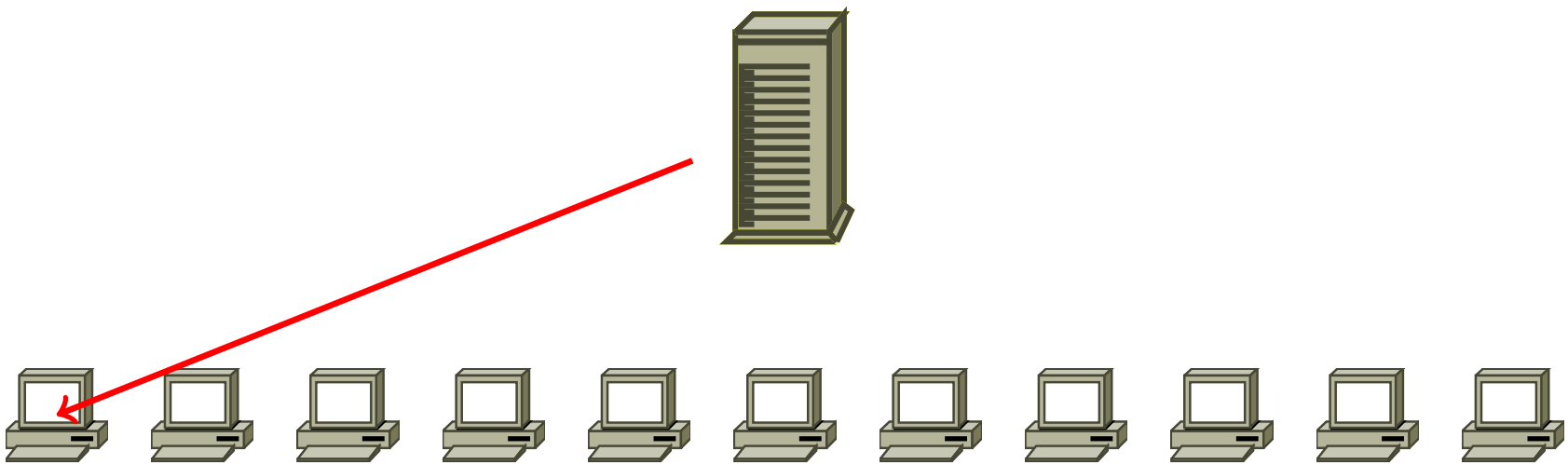
Scalable remote command execution with Taktuk

- **Goal:** execute commands on a large number of nodes



Scalable remote command execution with Taktuk

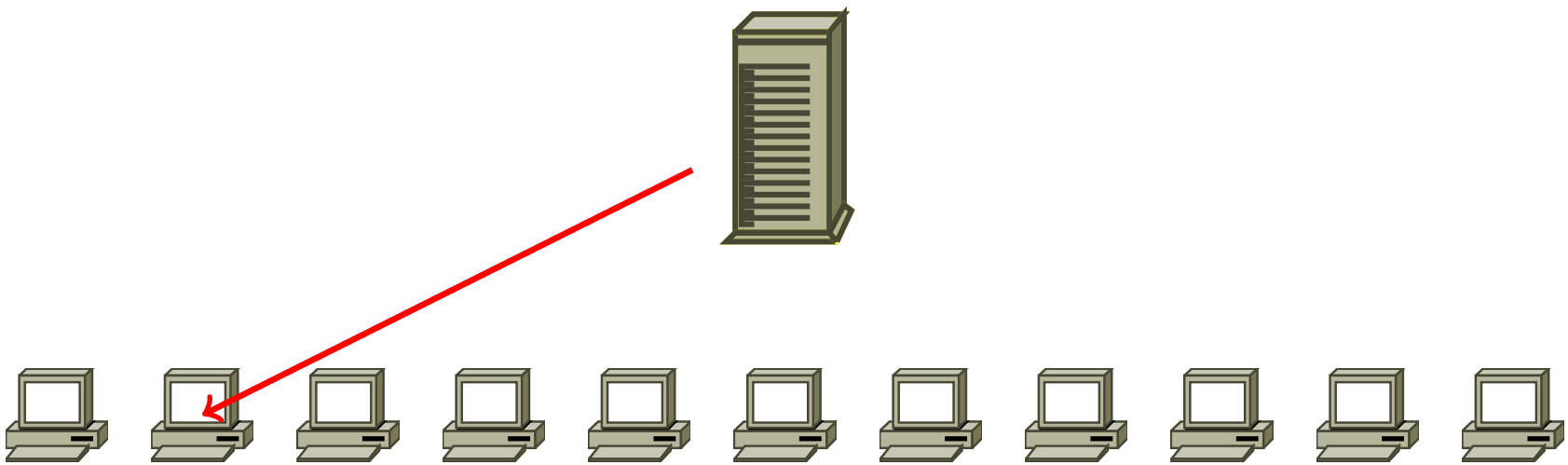
- **Goal:** execute commands on a large number of nodes



Sequential?

Scalable remote command execution with Taktuk

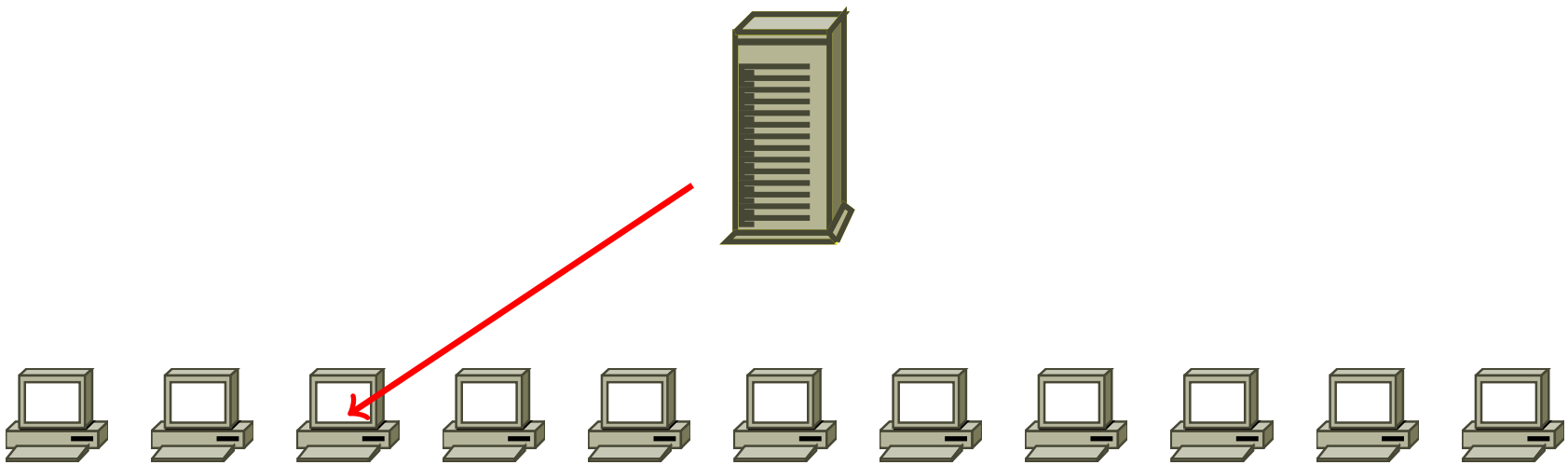
- **Goal:** execute commands on a large number of nodes



Sequential?

Scalable remote command execution with Taktuk

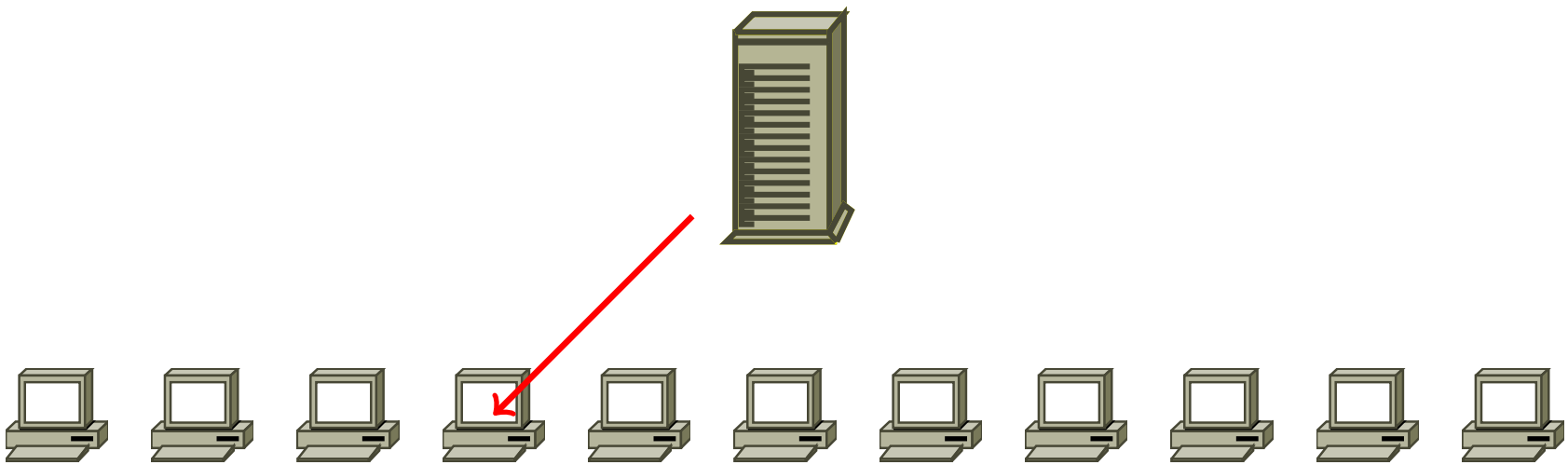
- **Goal:** execute commands on a large number of nodes



Sequential?

Scalable remote command execution with Taktuk

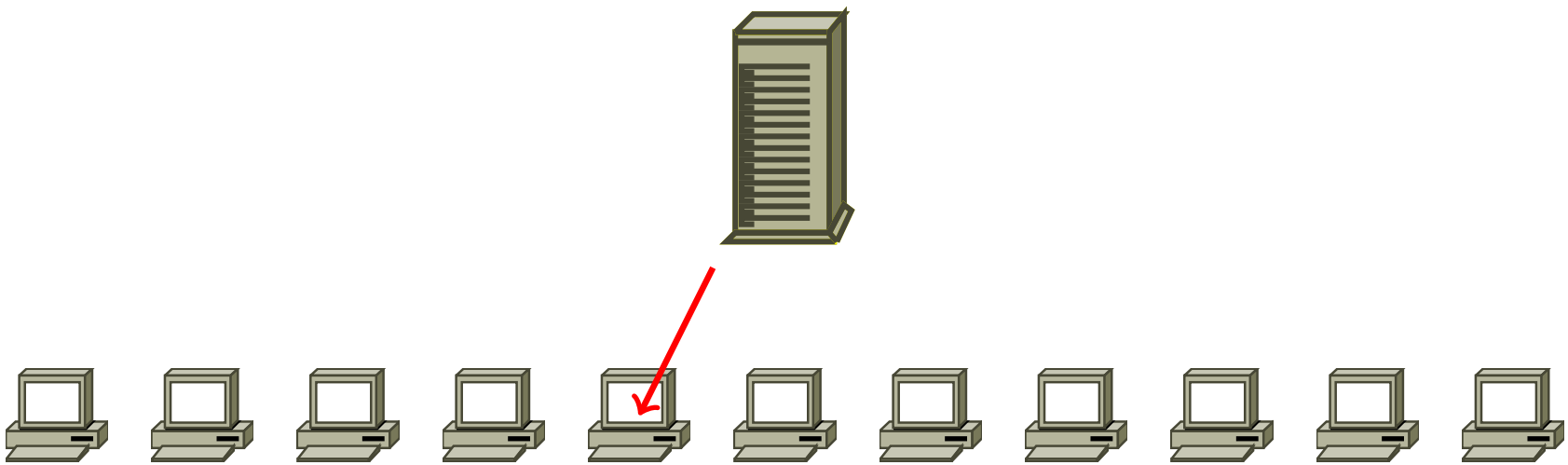
- **Goal:** execute commands on a large number of nodes



Sequential?

Scalable remote command execution with Taktuk

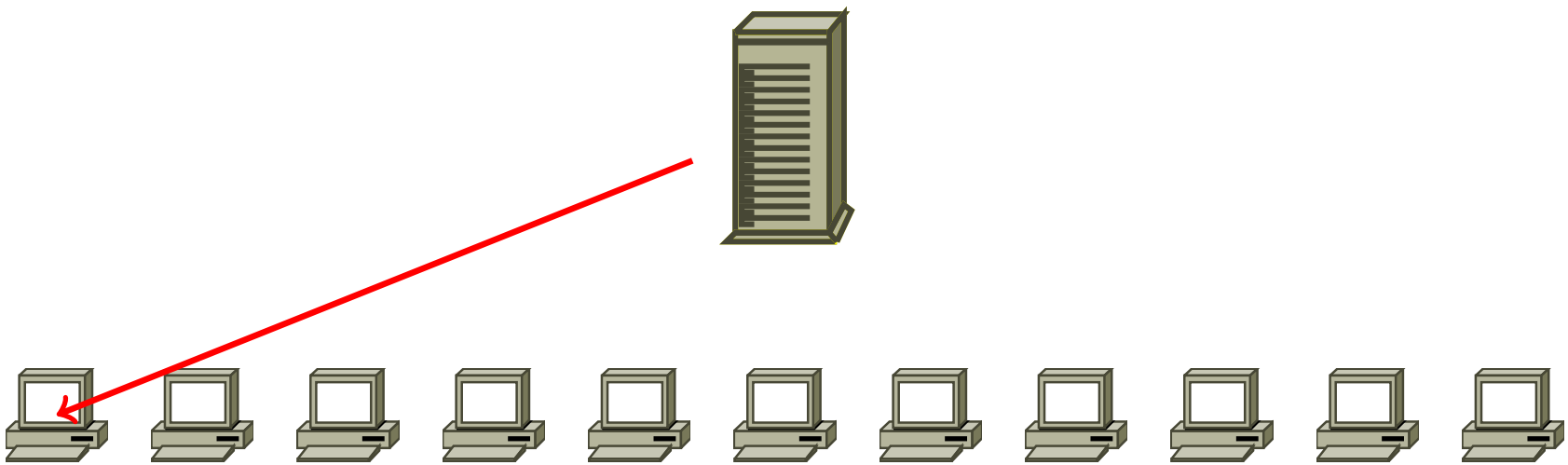
- **Goal:** execute commands on a large number of nodes



Sequential?

Scalable remote command execution with Taktuk

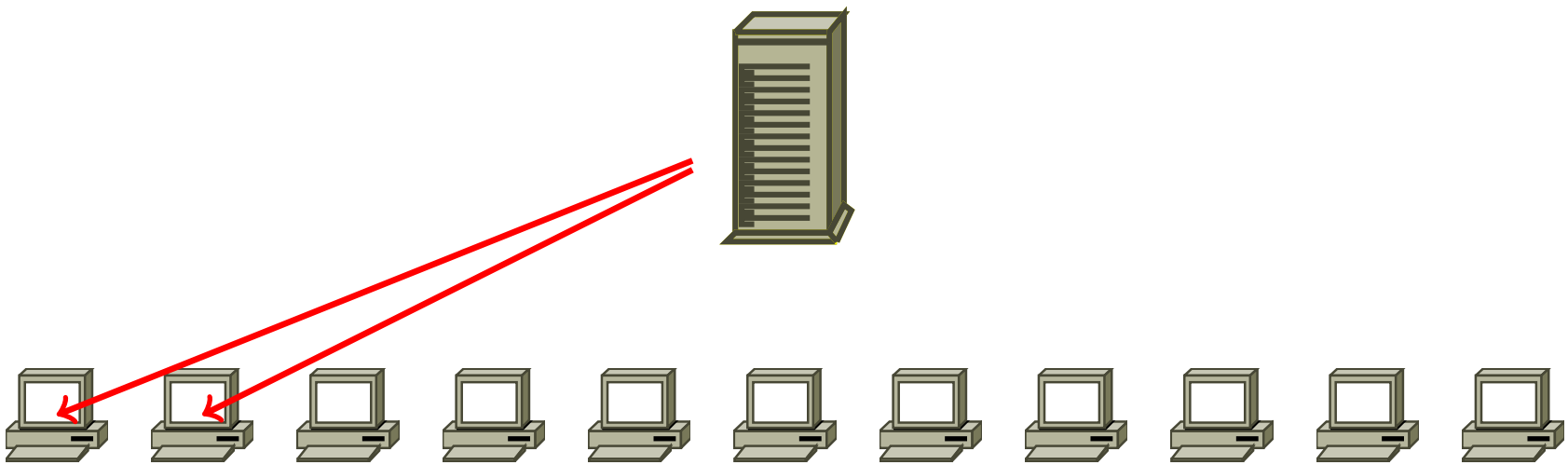
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

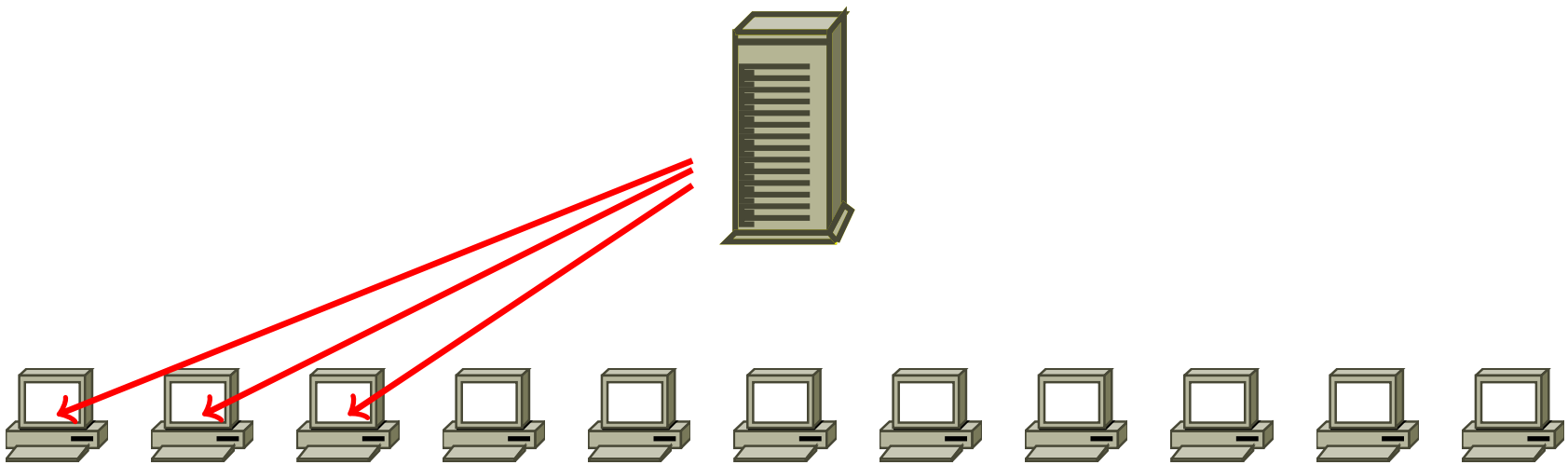
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

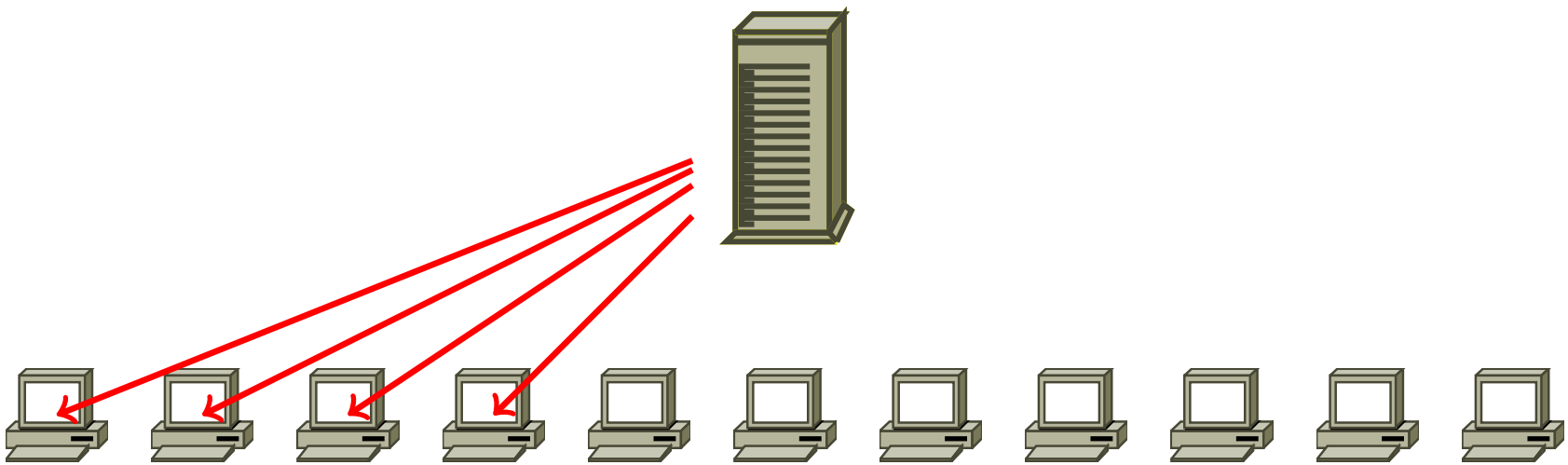
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

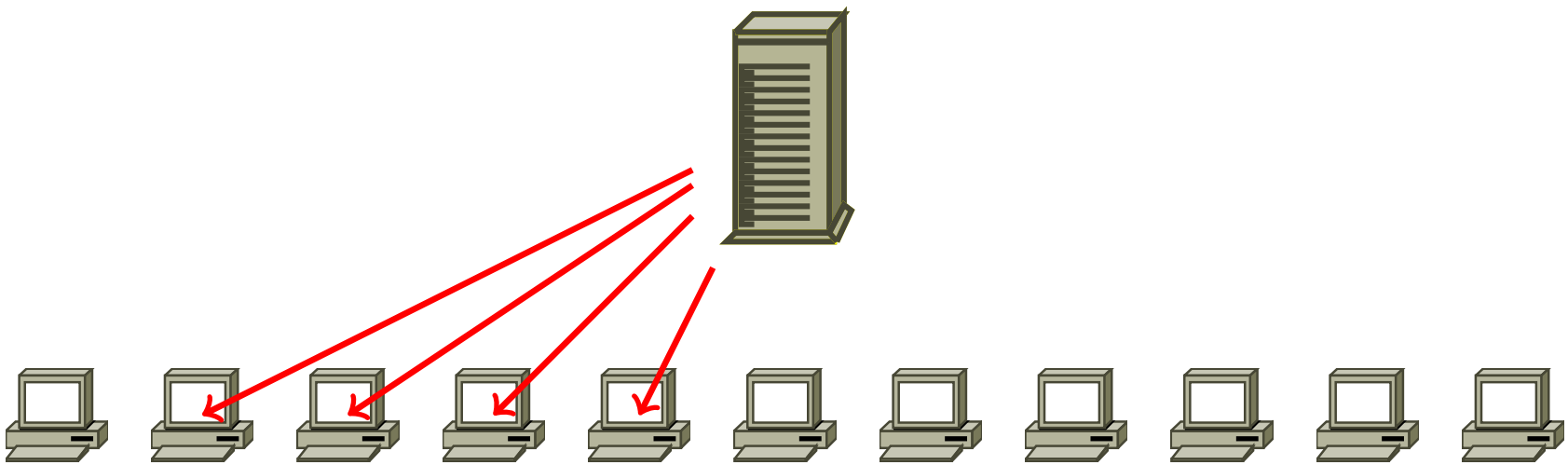
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

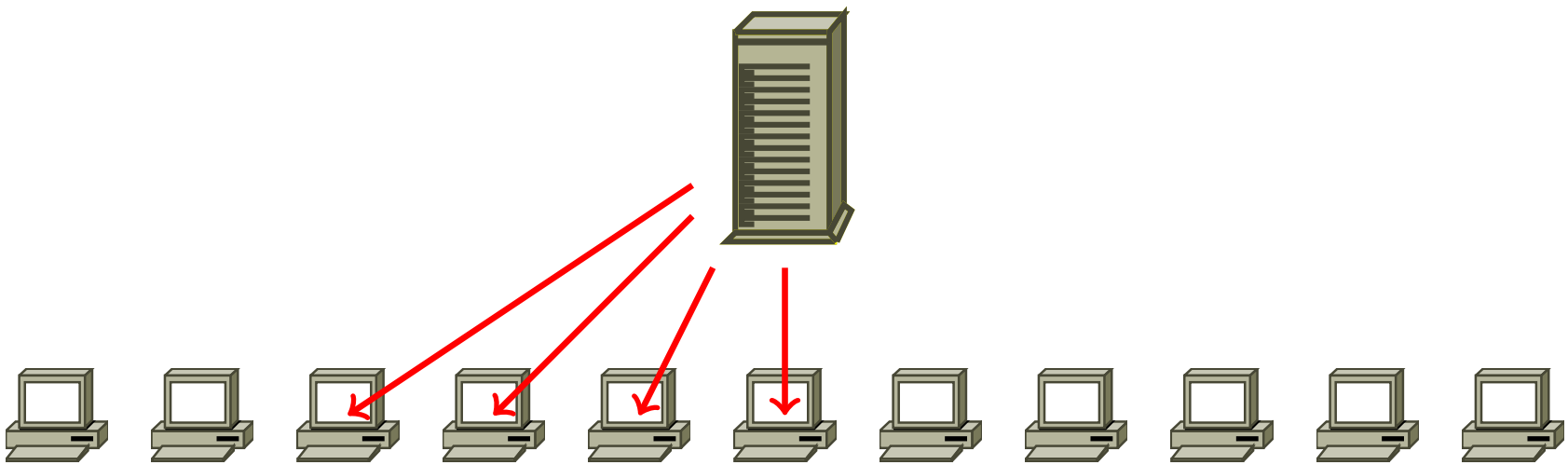
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

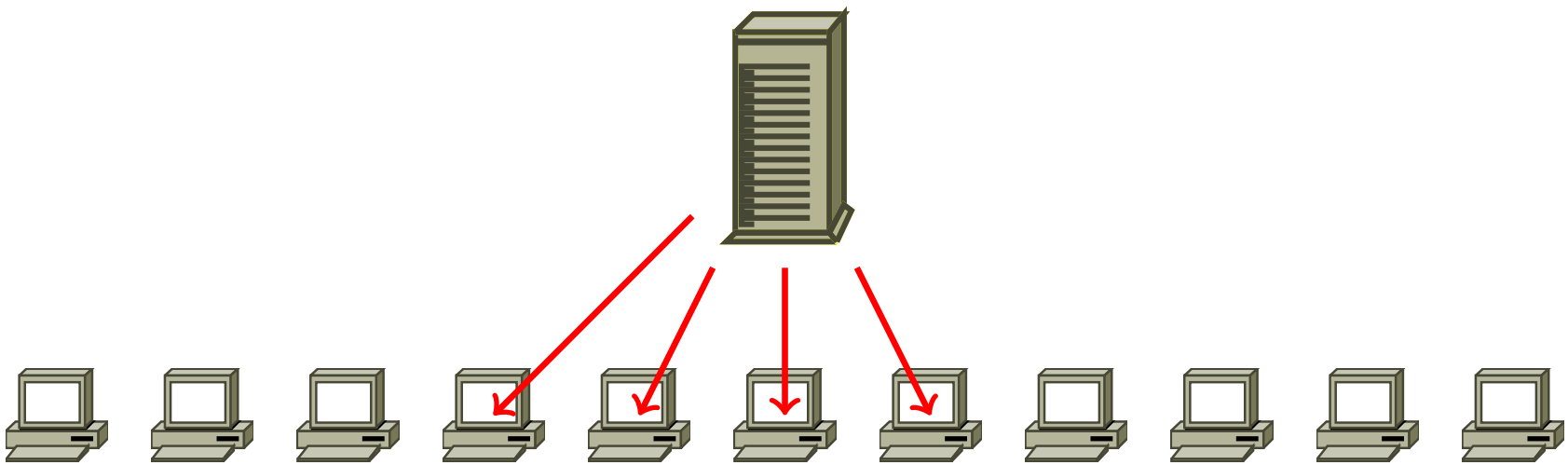
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

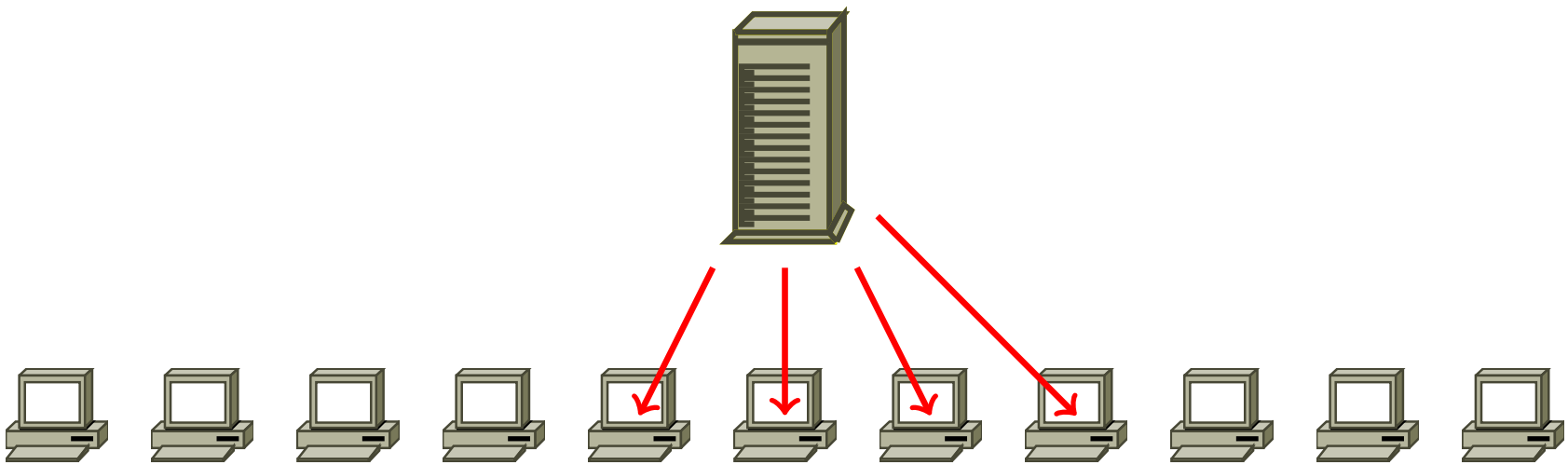
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

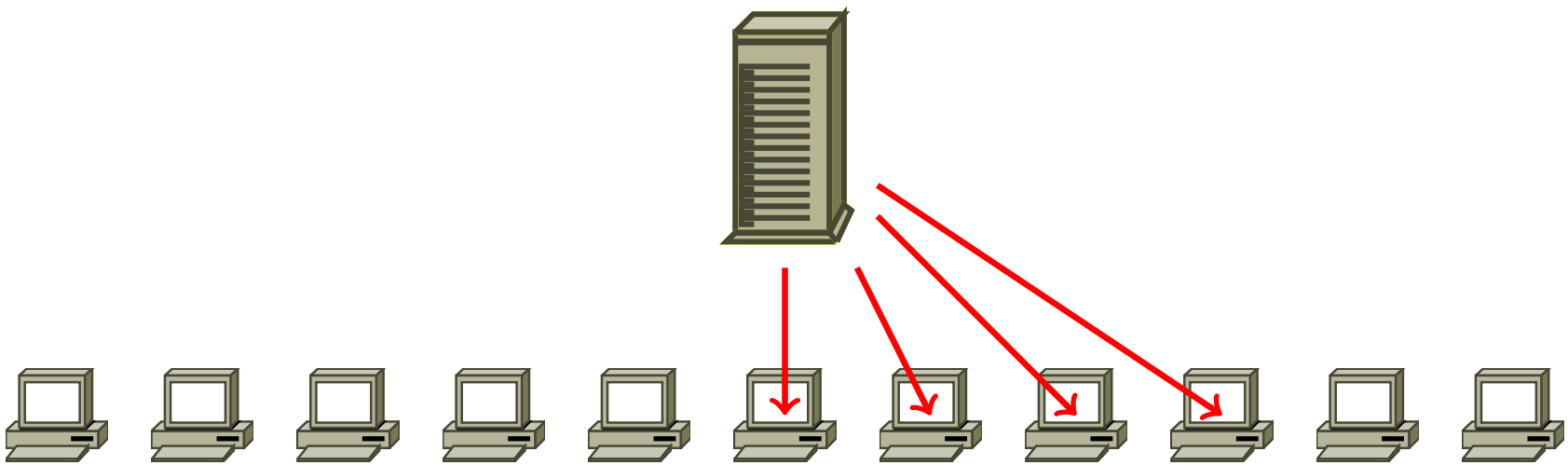
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

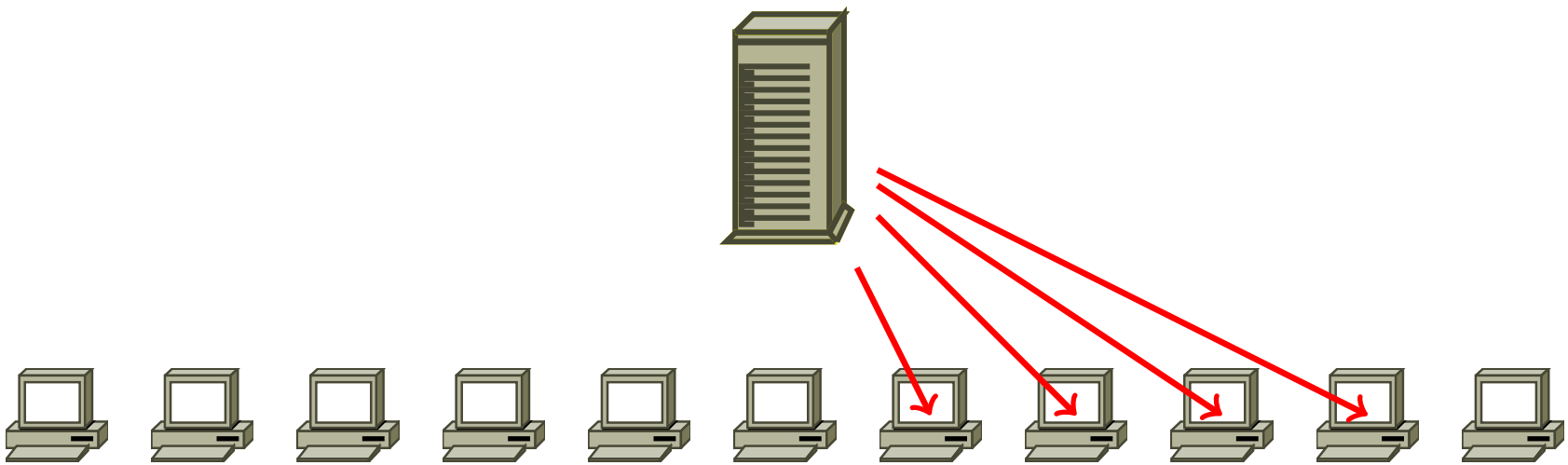
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

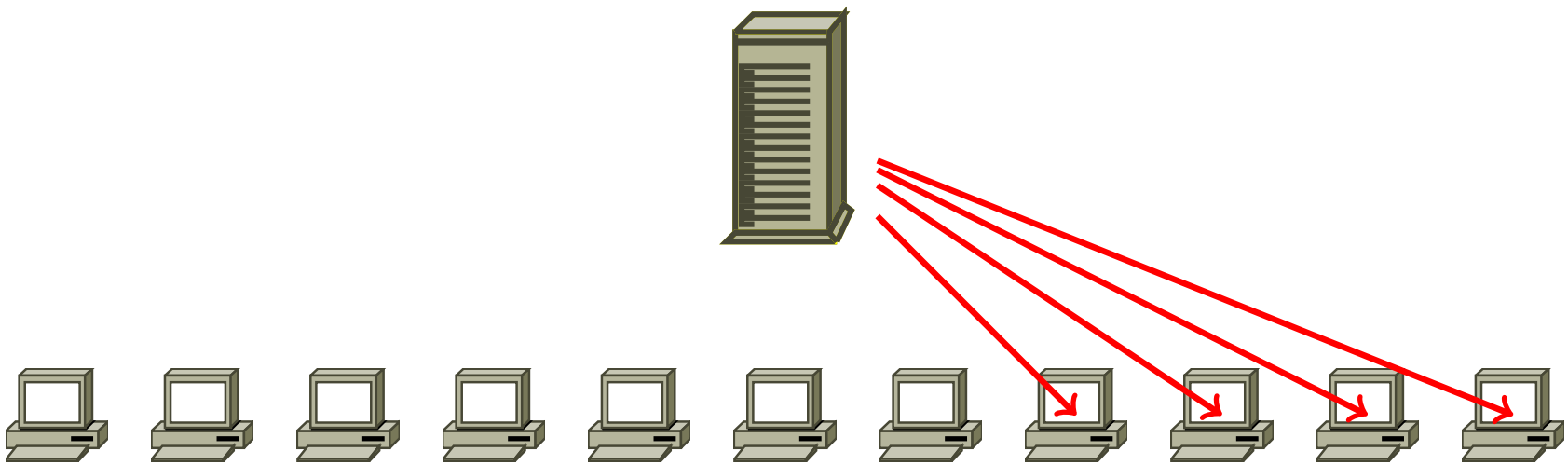
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

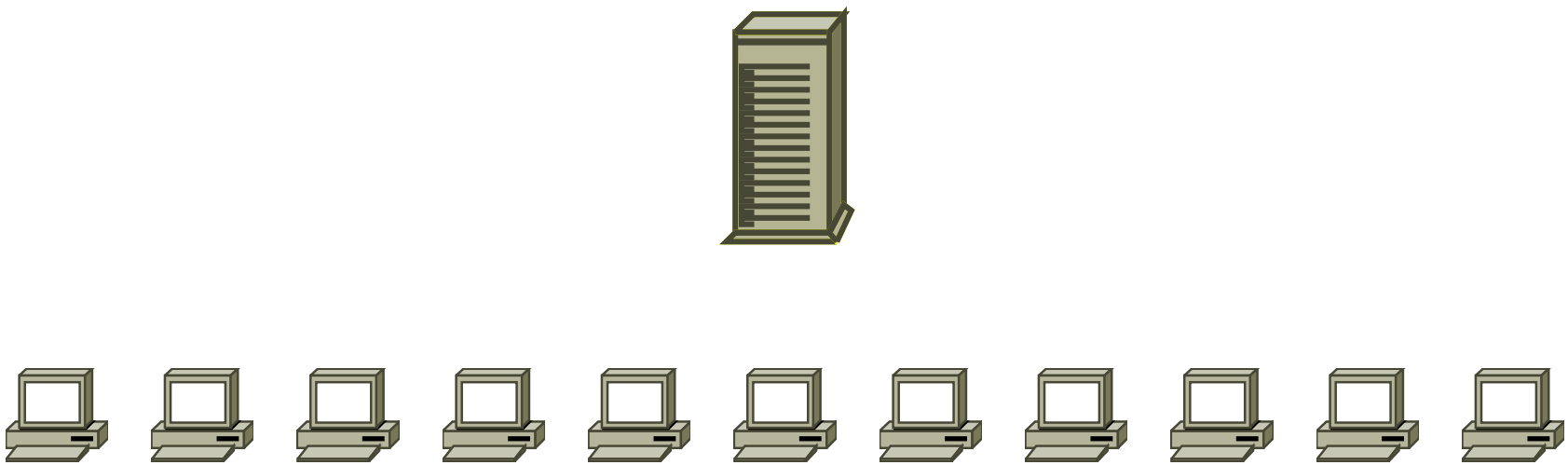
- **Goal:** execute commands on a large number of nodes



Sequential + sliding window (pdsh)?

Scalable remote command execution with Taktuk

- **Goal:** execute commands on a large number of nodes

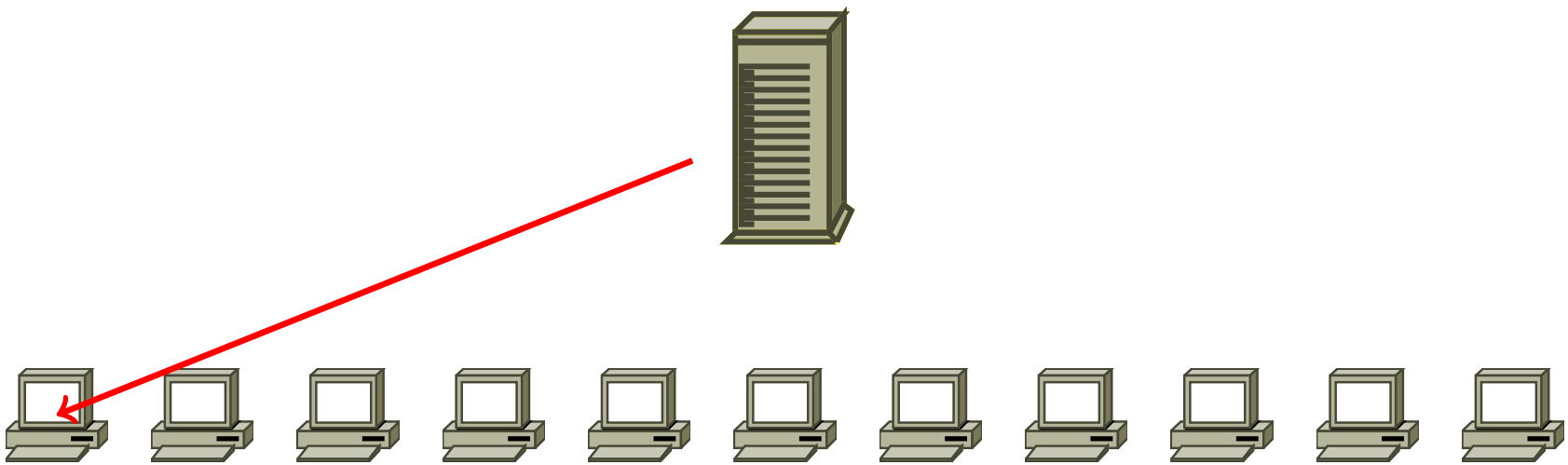


In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- 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

- **Goal:** execute commands on a large number of nodes

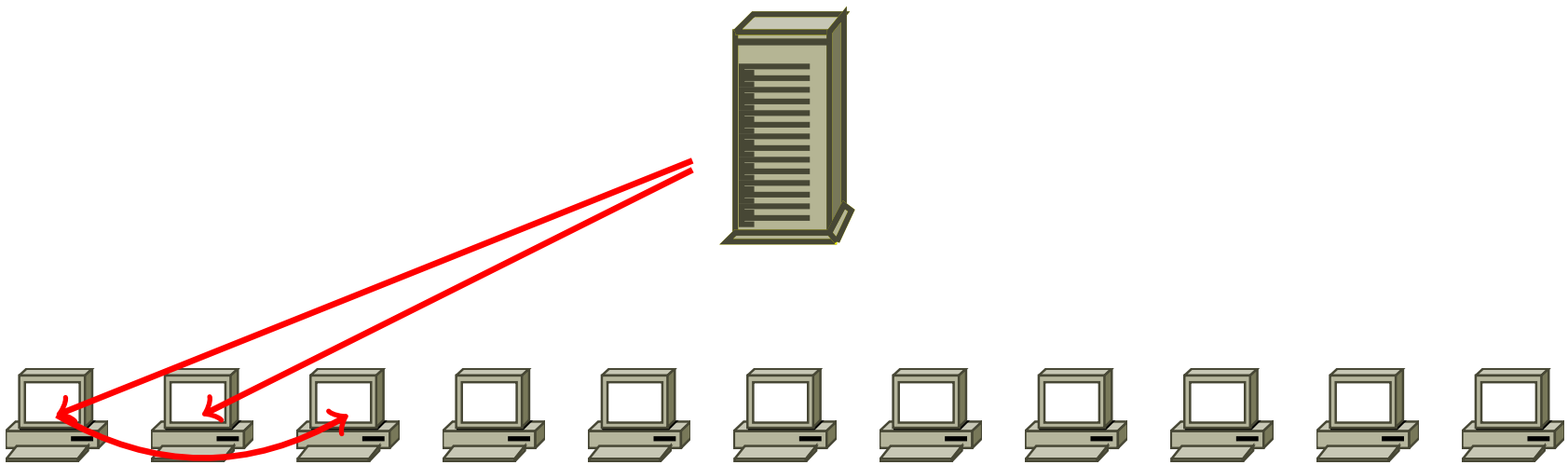


In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- 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

- **Goal:** execute commands on a large number of nodes

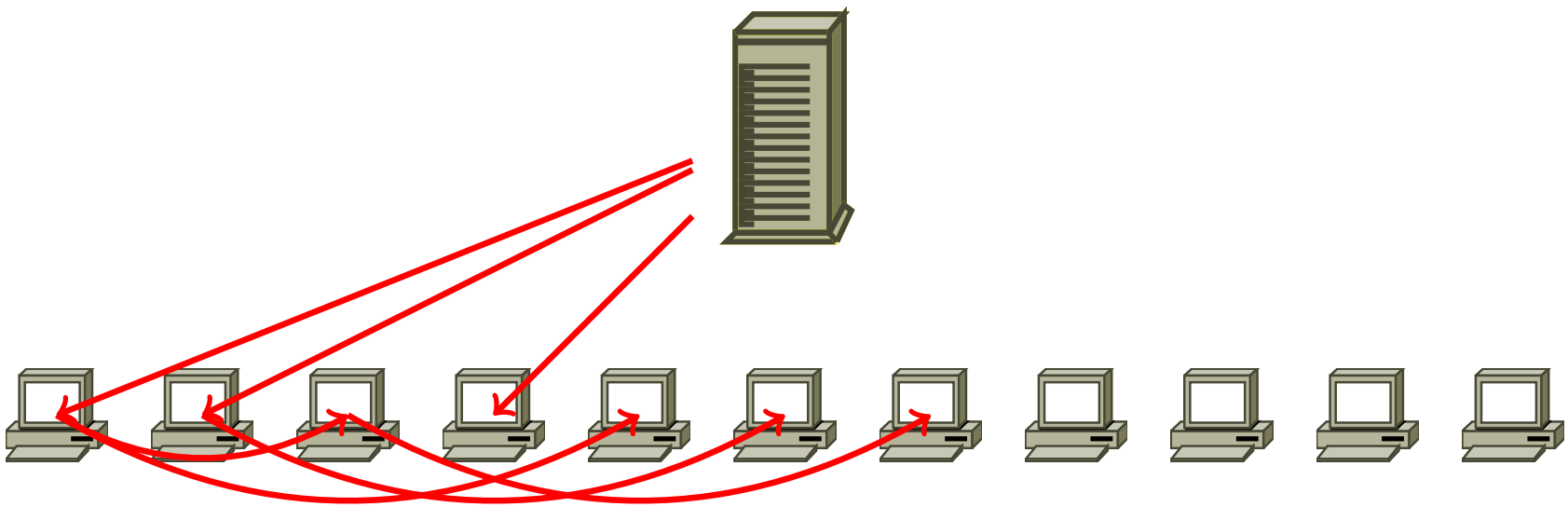


In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- 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

- **Goal:** execute commands on a large number of nodes

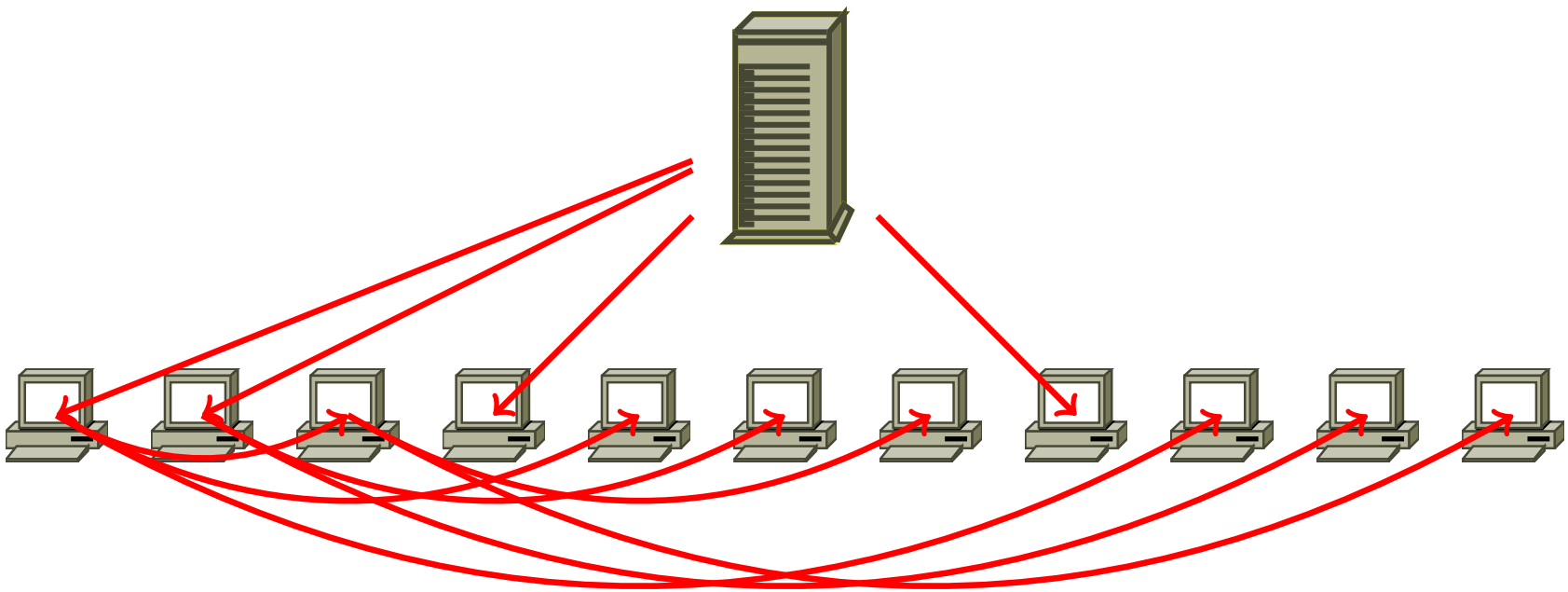


In Kadeploy: **Tree-based** $\rightsquigarrow$ logarithmic complexity (vs linear)

- 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

- **Goal:** execute commands on a large number of nodes



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.

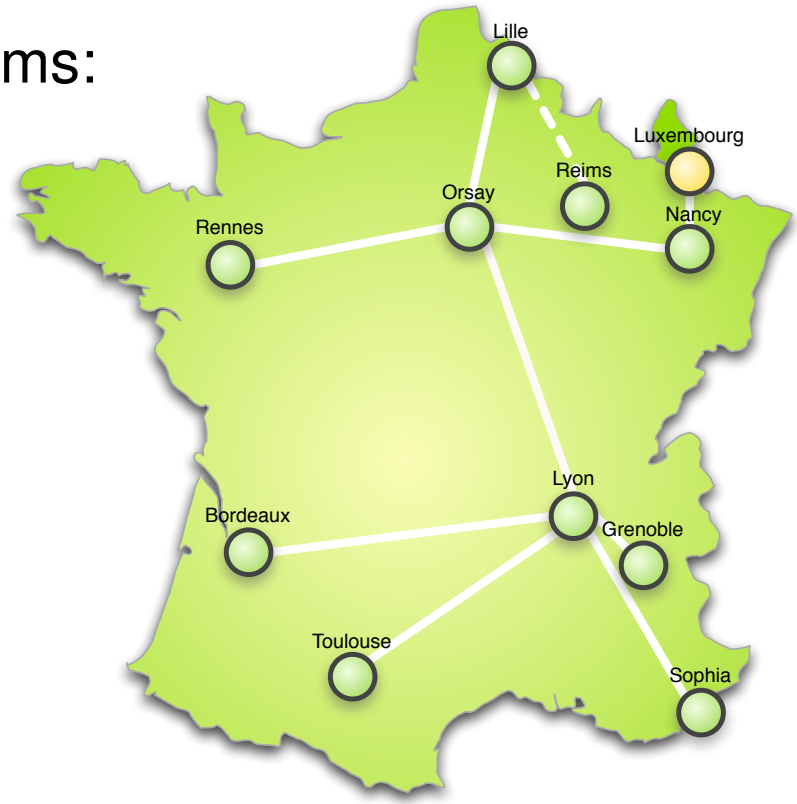
Grid'5000

Testbed for research on distributed systems:

- High Performance Computing
- Cloud computing
- Peer-to-peer systems
- Grids

Current status:

- 11 sites (1 outside France)
- 26 clusters
- 1300 nodes
- 8000 cores
- Dedicated 10 Gbps network
- Diverse technologies:
  - Intel (60%), AMD (40%)
  - CPUs from one to 12 cores
  - Myrinet, Infiniband {S,D,Q}DR
  - Two GPU clusters, one MIC cluster
- 500+ users per year



Reconfiguring the testbed with Kadeploy

- Provides a *Hardware-as-a-Service* Cloud infrastructure
- Enable users to deploy their own software stack & get *root* access
- Standard environments provided to users
  - Various GNU/Linux distribution
  - Automated deployment of Cloud stacks (OpenStack)

Reconfiguring the testbed with Kadeploy

- Provides a *Hardware-as-a-Service* Cloud infrastructure
- Enable users to deploy their own software stack & get *root* access
- Standard environments provided to users
  - Various GNU/Linux distribution
  - Automated deployment of Cloud stacks (OpenStack)
- Integrated with KaVLAN – Network isolation by reconfiguring switches for the duration of a user experiment
  - Avoid network pollution (broadcast, unsolicited connections)
  - Enable users to start their own DHCP servers
  - Experiment on ethernet-based protocols
  - Interconnect nodes with another testbed without compromising the security of Grid'5000

Using Grid'5000 and Kadeploy to test Kadeploy

- Used Grid'5000, Kadeploy and KaVLAN to create a *Cloud* of virtual machines
  - 4000 virtual machines
  - On 668 physical machines
  - From 4 sites of the Grid'5000 testbed
  - In a single L2 network spanning 1000 km
- Installed those virtual machines using Kadeploy

Questions?

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

http://www.grid5000.fr/

(Open Access program available:
https://www.grid5000.fr/open-access)

lucas.nussbaum@loria.fr