Efficient and Scalable OS Provisioning with Kadeploy3

Emmanuel Jeanvoine, Luc Sarzyniec and Lucas Nussbaum

Key features

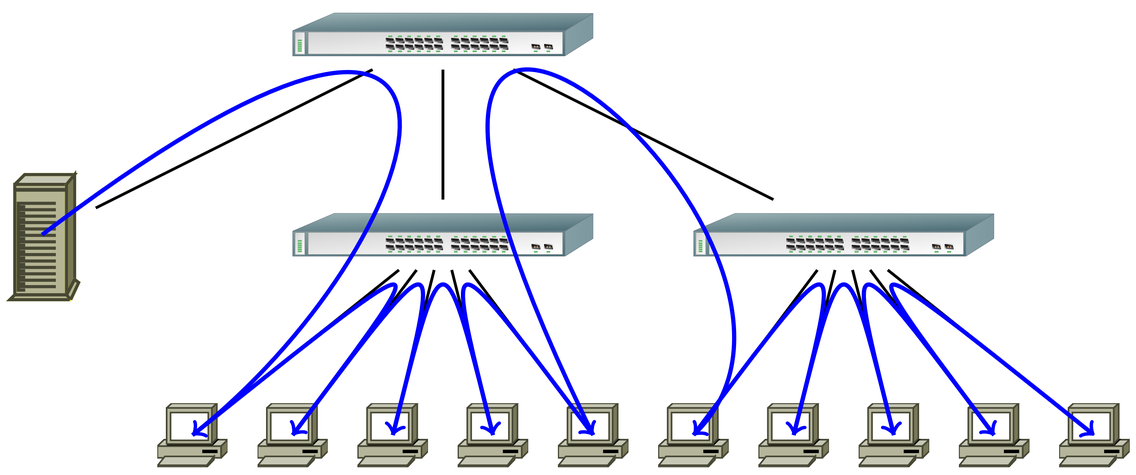
- **Install and configure a large number of nodes**
  - Install several cluster in one shot
  - Support for concurrent deployments
  - Control several clusters from a single client
- **Manage a library of pre-configured system images**
  - User-provided images
  - Visibility of images (shared, private)
- **Reliability of the installation process**
  - Customizable workflow engine
  - Windowed operations
  - Escalation of low-level remote commands
- **Hardware compatibility**
  - Built on top of PXE, DHCP, TFTP/HTTP
  - Remote operations based on SSH
  - Customizable remote low-level operations (IPMI, ...)
- **Software compatibility**
  - Support any operating system (Linux, *BSD, Windows, ...)
  - Integration with batch scheduler and network isolation tools
  - Remote control API

Scalability

System image broadcast

Goal: Send a big file on thousands of nodes
Challenge: Avoid network bottlenecks, saturation of links

- **Several alternatives available**
  - Chain, Tree, Bittorrent, ...



- Default alternative: **Topology-aware chain broadcast**
  - Parallel tree-based initialization of the chain
  - **Saturation of full-duplex** network in both directions
  - Efficient on networks composed of **hierarchy of switches**

Parallel operations

Goal: Executing commands on thousands of nodes
Challenge: Avoid client overloading, gather commands outputs

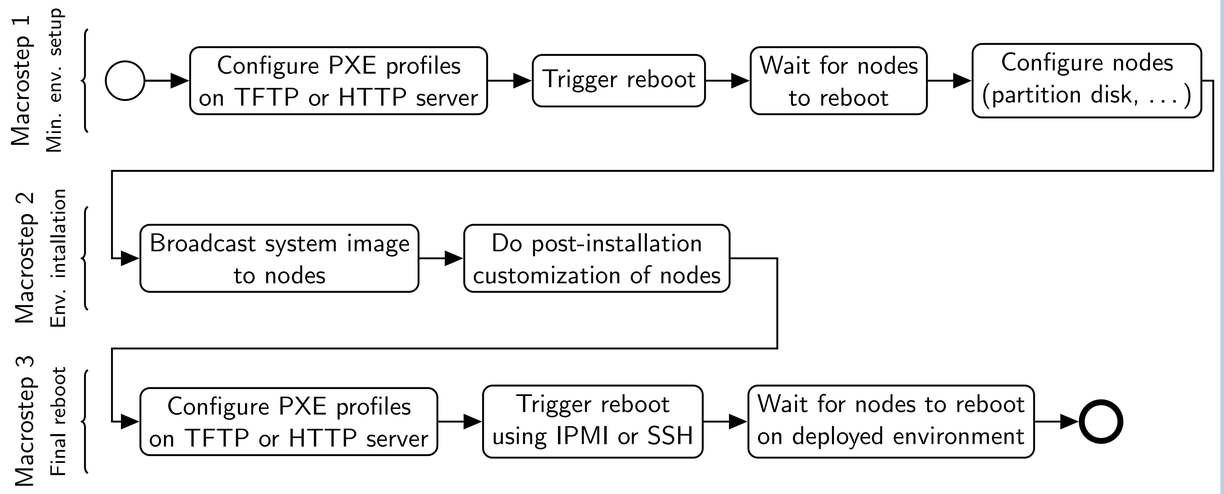
- Based on **TakTuk** (`http://taktuk.gforge.inria.fr`)
- **Hierarchical connections** between nodes
- Adaptative work-stealing algorithm
- **Auto-propagation** mechanism

Reliability

Reliable workflow engine

Goal: Manage the installation process
Challenge: Handle hardware & network failures, customization

- Engine based on **event automata**
- **Fallback methods** in case of failure
- Timeouts and retries at every step
- A typical workflow example:



Reliable reboot and power operations

Goal: Trigger remote reboot and power on/off on nodes
Challenge: Reliability, compatibility

- **Compatibility** with remote hardware managements protocols
- **Escalation** of several level of administrator defined commands
- Managing groups of nodes (e.g. PDU reboots)
- **Windowed operations** (DHCP flood, electrical hazards, ...)

Evaluation

- Key software on Grid'5000 **since 2009**
  - **25 clusters on 10 sites**
  - 620 users, **170 000 deployments**
  - about **10 mins** to deploy 130 nodes
- Virtualized infrastructure
  - **4000 VMs** dispatched on **635 physical nodes**
  - **3838 nodes** successful in a single shot in **less than 1 hour**

Software suite

- **Management of images**
  - User custom images
- **Rights management**
  - Compatibility with batch scheduler
- **Statistics collection**
  - Identify hardware issues, ...
- **Frontends to low-level tools**
  - Reboot and power on/off operations, serial consoles
- **DEB** and **RPM** packages
- Actively developed since 2009

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

Grid'5000 — informatics mathematics Inria — Université de Lorraine — Loria Laboratoire lorrain de recherche en informatique et ses applications — jres Montpellier 2013