

Automatic Testing Infrastructure for OSCAR

Benoît des Ligneris, Ph. D
Revolution Linux, president
benoit.des.ligneris@revolutionlinux.com
<http://www.revolutionlinux.com/>



Plan

- Rationale
- Available technologies
- Proposed implementation
- Proposed test plan
- Conclusion

Rationale

- Testing OSCAR is a $O(n)$ problem where n is the number of supported {distribution,version,architecture}
- As OSCAR become more and more popular (new distribution, new architecture) and because the existing distributions release 2 or 3 version a year, releasing an official OSCAR version consumes a lot of developper time

Rationale / Future

- OSCAR supports now experimentally Debian (on IA64), Mandrake (on x86_64), white-box (on IA64), etc.
- Each change to the code should be tested on all {distribution,version,architecture}. This is not possible because it will slow the developing process too much !



Available technologies

- Virtualization of computers is now a « common » technology
 - Commercial : vmware
 - www.vmware.com
 - Emulate linux on linux : plex86
 - www.plex86.org
 - Emulate a i386 CPU : bochs
 - <http://bochs.sourceforge.net/>



Avalaible technologies

- Linux inside Linux :
 - The chroot method :
 - man chroot
 - The Linux vservers
 - www.linux-vservers.org
 - User Mode Linux (UML Linux)
 - <http://user-mode-linux.sourceforge.net/>



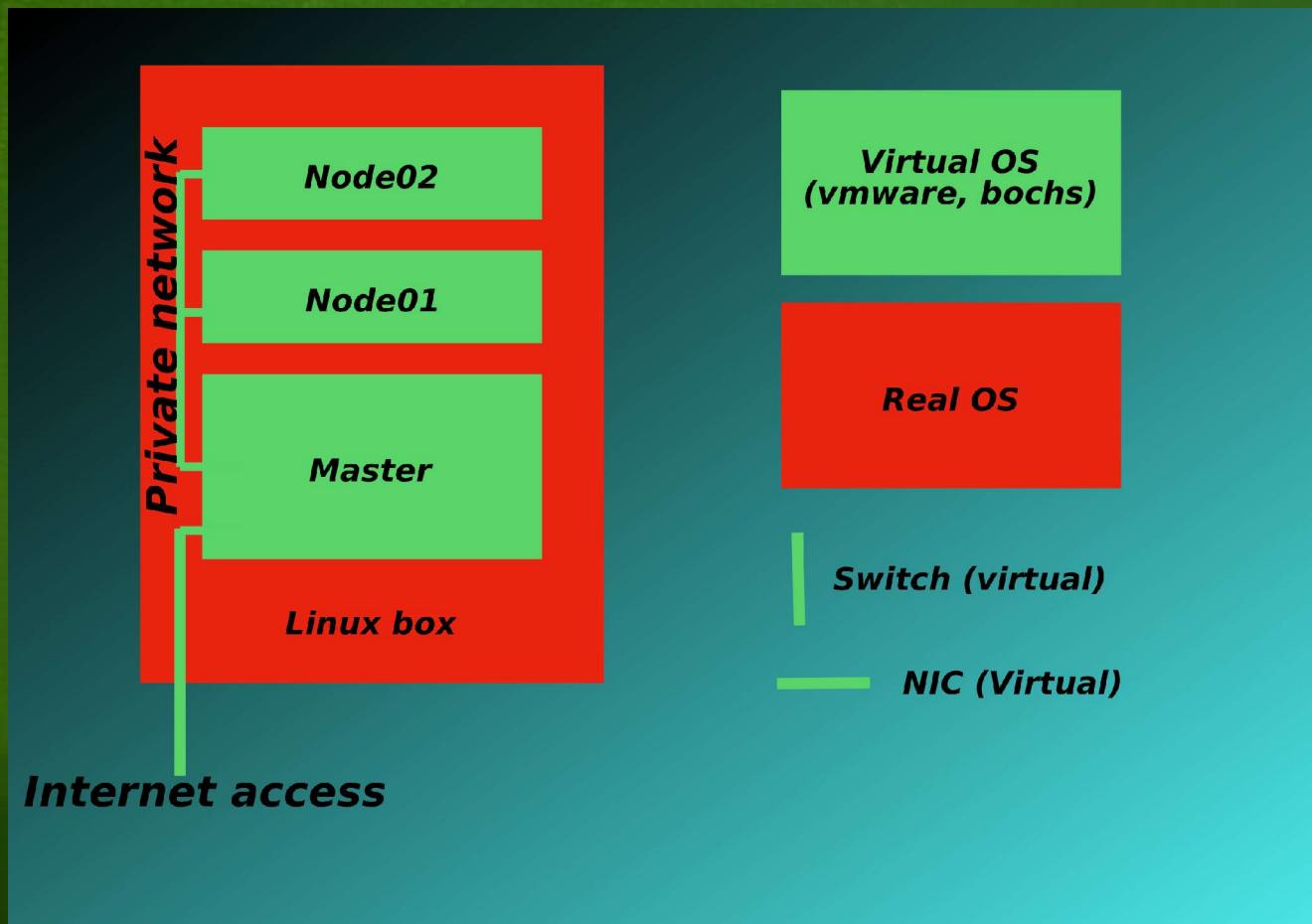
Recommended technology

- Vmware :
 - + : already used by OSCAR developers
 - + : no kernel changes required (guest OS or host OS)
 - + : SCSI/IDE support
 - - : only support i386
 - - : cost 250\$/licence

Architecture independance

- Two architecture independant solutions :
 - UML Linux (kernel + user patches)
 - Chroot
 - Can only be used to test already installed computers
 - Problems with hardware « sharing » on the same host
- UML required some changes to the host OS

Network Topology of the testing infrastructure

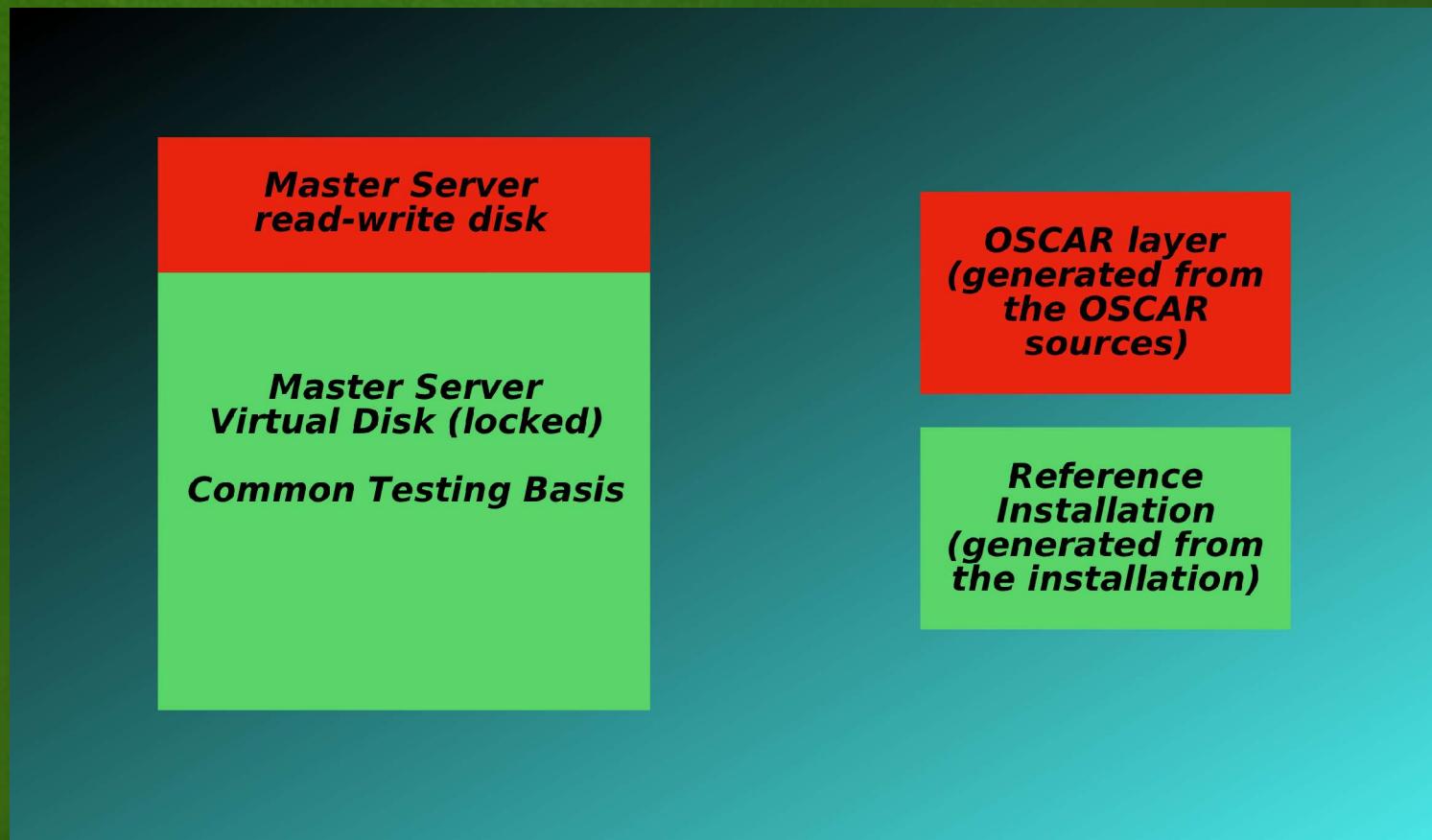


Proposed implementation

- The starting point of the test will be an already installed distribution (RedHat, Fedora, Mandrake, Debian, etc.)
- This is called the « initial state » of the testing process. This initial state will be publicly available and shared across developpers.
- Then OSCAR installation process will be tested.



Hard disk structure



Proposed test plan

- Several levels of functionnality can be tested :
 - (1) OS install / performance
 - (2) OSCAR master-node install / performance
 - (3) OSCAR Node installation / performance
 - (4) OSCAR Package installation / usage / performance
 - (5) Package sets installation / usage / performance
 - (6) OSCAR maintenance functions

Master-node install / performance

- Install OSCAR on the virtual master node :
 - Only core packages
- Test for functionality :
 - Core packages tests
- Measure efficiency

Node installation / performance

- Use the OSCAR tools (SIS) to remotely and automatically install two nodes
 - Test for cluster functionality
 - Measure performance (installation time)
 - Test all possible options (for instance multicast, 1 NIC, no internet access, etc.)



Package installation / usage / performance

- Each package (+ its dependency) will be installed and then tested indepedently
 - Installation possible (yes/no)
 - Installation performance (time)
 - Installation quality (tests provided by the package author)

Package sets installation / usage / performance

- Because packages interacts with each other, it is necessary to test package sets :
 - Installation possible (yes/no)
 - Installation performance (time)
 - Installation quality (tests provided by the package-set author)

Maintenance functions

- The basic maintenance operations of OSCAR will be tested :
 - add/remove node
 - add/remove package
 - push/sync image

Conclusion

- Available technology reviewed
- Vmware easier but expensive, work on only one architecture
- chroot : more interesting but need more work ?
- Different level of testing possible :
 - OS level, OSCAR level, node, package, package sets and OSCAR maintenance tools