Unattended System Deployment for Cloud and Big Data Computing Environment

Steven Shiau, Robert Lin, Thomas Tsai,

Sakana Huang, Ceasar Sun

clonezilla.org

Q4, 2016

Outline

- Introduction to Clonezilla
  - Features
  - Updates since 2016 Spring
- Unattended system deployment
  - Live boot parameters
  - Ansible
- Big data system deployment
  - Clonezilla-BD
- Q&A

Outline

- Introduction to Clonezilla
  - Features
  - Updates since 2016 Spring
- Unattended system deployment
  - Live boot parameters
  - Ansible
- Big data system deployment
  - Clonezilla-BD
- Q&A

System imaging and cloning - backup

image source: maggiesfarm.anotherdotcom.com
www.compsults.com, and jervisdabreo.com

Massive system deployment

About us

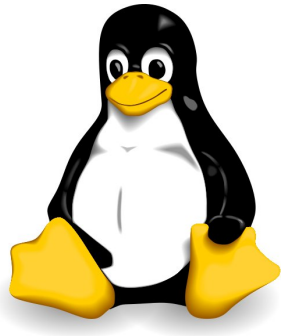
- Developers of the free software DRBL, Clonezilla and more...
- Steven is also the maintainer of GParted live CD
- From Taiwan, working for the NPO NCHC (National Center for High-Performance Computing)

財團法人國家實驗研究院
國家高速網路與計算中心
National Center for High-Performance Computing
Better HPC Better Living

Taiwan image source: wikipedia.org

What is Clonezilla?

- A partition and disk imaging/cloning utility similar to True image® or Ghost®
- GPL license
- A bare metal recovery tool for


*1


*2

*3


*4

VMFS

VMware ESX/ESXi

*5

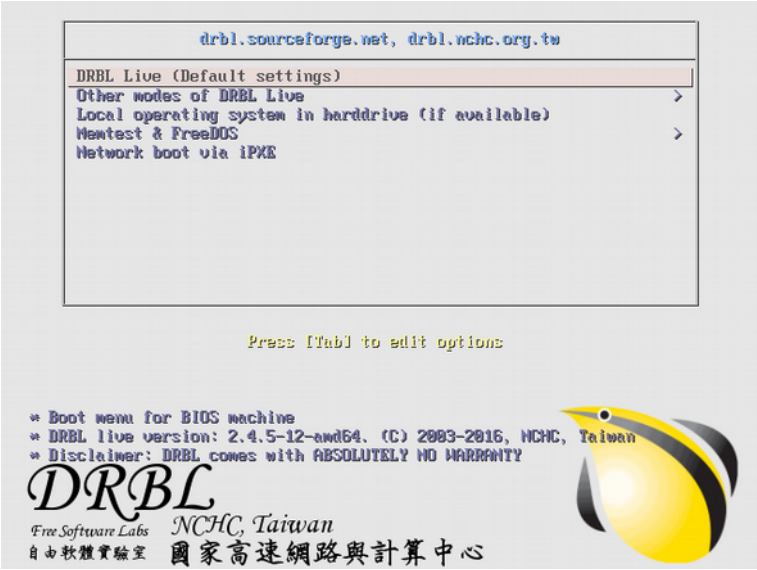
*6

*Logo source: (1) Larry Ewing, Simon Budig and Anja Gerwinski, (2) Apple ,(3) Microsoft, (4) Marshall Kirk McKusick, (5) VMWare (6) Distrowatch.com

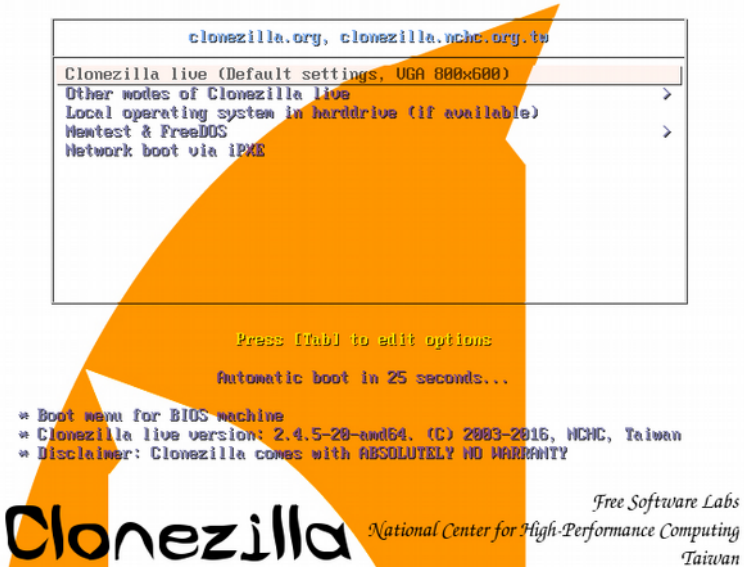
Clonezilla Features

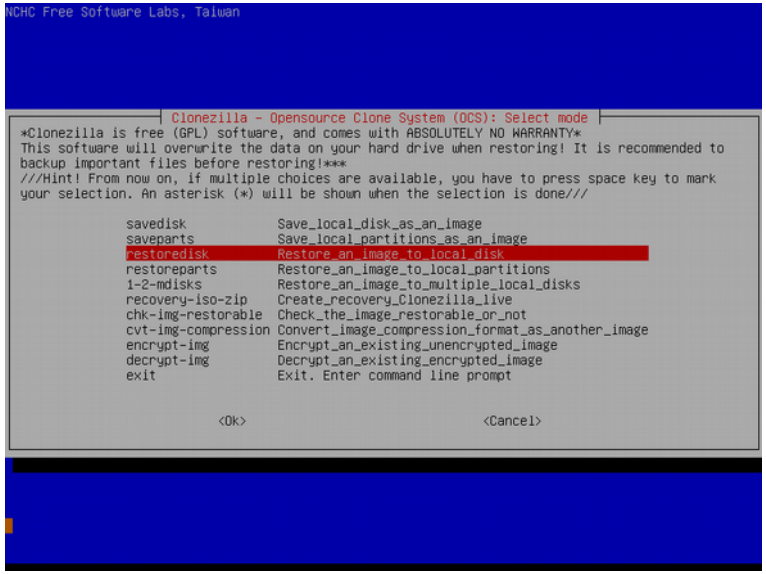
- Free (GPL) Software
- File systems supported:
  - Ext2/3/4, ReiserFS, Reiser4, XFS, JFS, HFS+, BrtFS, F2fs, UFS, Minix, VMFS, F2FS, NILFS2, FAT and NTFS
  - Supports LVM2
  - Support some hardware RAID chips (by kernel)
- Smart copying for supported filesystem. For unsupported file systems sector-to-sector copying is done via dd.
- Boot loader : syslinux, grub 1/2 ; MBR and hidden data (if exist)
- Serial console
- Unattended mode
- One image restoring to multiple local devices
- Multicast supported in Clonezilla Server Edition (SE)
- The image format is transparent, open and flexible

DRBL live,
i.e. Clonezilla Server Edition



Clonezilla Live





Developers

- Steven Shiau
- K. L. Huang
- Ceasar Sun
- Jazz Wang
- Thomas Tsai
- Jean-Francois Nifenecker
- Louie Chen
- Nagappan Alagappan







Language file contributors

- English (en_US): Dylan Pack.
- German (de_DE): Michael Vinzenz.
- Hungarian (hu_HU): Greg Marki
- Spanish (es_ES): Juan Ramón Martínez and Alex Ibáñez López.
- French (fr_FR): Jean-Francois Nifenecker and Jean Francois Martinez.
- Italian (it_IT): Gianfranco Gentili.
- Japanese (ja_JP): Akira Yoshiyama and Annie Wei.
- Brazilian Portuguese (pt_BR): Marcos Pereira da Silva Cruz.
- Russian (ru_RU): Anton Pryadko and Igor Melnikov.
- Slovak (sk_SK): Ondrej Dzivy Balucha
- Turkish (tr_TR): Ömer YILDIZ
- Simplified Chinese (zh_CN): Zhiqiang Zhang and Liang Qi.
- Traditional Chinese (zh_TW): T. C. Lin.

Partners

- The following companies either embed Clonezilla in their products or promote Clonezilla:
  - Linmin
  - eRacks Open Source Systems
  - Miracle Linux

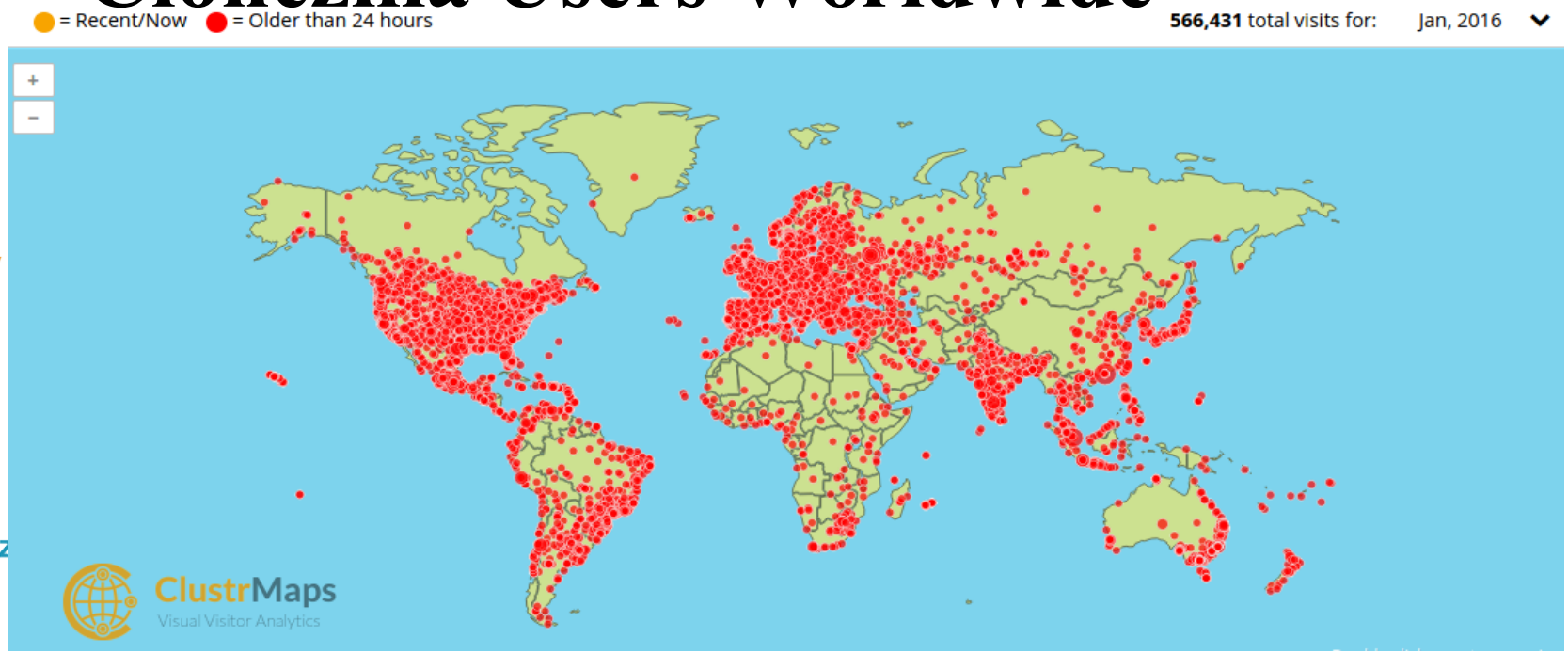
Changes and new features from 2016/Q1

- Inspect the files checksum in the partition. To use it, enable "-gmf" option in expert mode when saving an image and enable "-cmf" option when restoring image. For disk to disk clone, use "-cmf" option.
- Support /dev/nbd device
- Add supporting for grub on EBR (Extended Boot Record) imaging and cloning.
- When mounting image repository, it is able to browse the directories recursively.
- Add support for boot parameter ocs_preload*. It can be used to fetch tarall/zip/sh files from http(s), ftp, tftp, and local URL then extract to /opt/.
- Image repository can be auto mounted with boot parameter ocs_repository in URI format, e.g. ocs_repository="dev:///dev/sdf1" or ocs_repository="smb://wa-domain;jack:mypass@192.168.7.25/images".

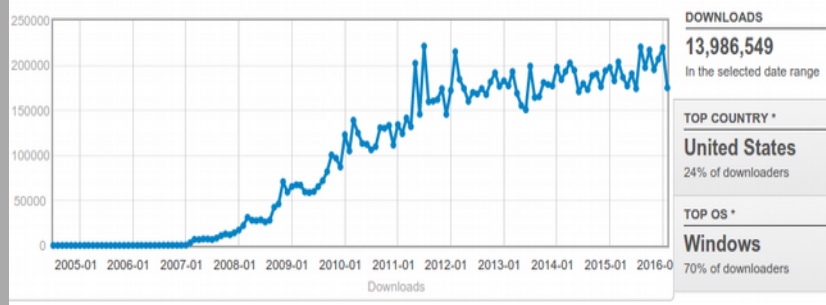
Changes and new features from 2016/Q1 (Continued)

- Add entry points by using boot parameters ocs_savedisk_prerun, ocs_saveparts_prerun, ocs_restoredisk_prerun, and ocs_restoreparts_prerun
- /opt/overwrite-all-boot-param, /opt/overwrite-part-boot-param
  - Downloaded from root of ocs_preload or other ways
  - “overwrite-all-boot-param” -> overwrite whole /proc/cmdline,
  - “overwrite-part-boot-param” -> only overwrite part of /proc/cmdline. Especially those "ocs_*" parameters.
- In summary, the boot parameters to be run in order:
  1. ocs_prerun* (might overwrite /proc/cmdline)
  2. ocs_preload
  3. ocs_repository
  4. ocs_savedisk_prerun/ocs_saveparts_prerun/ ocs_restoredisk_prerun/ocs_restoreparts_prerun
  5. ocs_postrun*

Clonezilla Users Worldwide




```
ca_ES.UTF-8 Catalan | Català
de_DE.UTF-8 German | Deutsch
en_US.UTF-8 English
hu_HU.UTF-8 Hungarian | Magyar
es_ES.UTF-8 Spanish | Español
fr_FR.UTF-8 French | Français
it_IT.UTF-8 Italian | Italiano
ja_JP.UTF-8 Japanese | 日本語
pt_BR.UTF-8 Brazilian Portuguese | Português do Brasil
ru_RU.UTF-8 Russian | Русский
sk_SK.UTF-8 Slovak | Slovenský
tr_TR.UTF-8 Turkish | Türkçe
zh_CN.UTF-8 Chinese (Simplified) | 简体中文
zh_TW.UTF-8 Chinese (Traditional) | 正體中文 - 臺灣
```



>13,000,000 downloads

Outline

- Introduction to Clonezilla
  - Features
  - Updates since 2016 Spring
- Unattended system deployment
  - Live boot parameters
  - Ansible
- Big data system deployment
  - Clonezilla-BD
- Q&A

Live boot parameters for unattended deployment

- Preset language and keyboard
  - locales=en_US.UTF-8 keyboard-layouts=us
- Remember the boot parameters to be run in this order:
  - ocs_prerun → ocs_preload → ocs_repository → ocs_savedisk_prerun/ocs_saveparts_prerun/ocs_restoredisk_prerun/ ocs_restoreparts_prerun → ocs_postrun
- E.g.
  - boot=live union=overlay username=user config components quiet noswap edd=on nomodeset locales=en_US.UTF-8 keyboard-layouts=us ocs_prerun1="dhclient -v eth0" ocs_repository="nfs://192.168.56.254:/home/partimag" ocs_live_run="ocs-sr -g auto -e1 auto -e2 --batch -r -j2 -scr -k1 -p true restoredisk xenial-x64-20161104 sda" ocs_live_extra_param="" ocs_live_batch="no" vga=788 ip= net.ifnames=0 nosplash i915.blacklist=yes radeonhd.blacklist=yes nouveau.blacklist=yes vmwgfx.enable_fbdev=1 ocs_postrun1="mount /dev/sda1 /mnt" ocs_postrun2="rm -f /mnt/etc/resolv.conf; echo nameserver 160.194.192.17 > /mnt/etc/resolv.conf" ocs_postrun3="chroot /mnt/ apt-get update; chroot /mnt/ apt-get -y install python" ocs_postrun4="reboot"

How about after the restored OS boots?

- Ansible
  - Free software from RedHat, http://ansible.com
- Required packages in the restored OS
  - Python
  - Ssh service
- For example
  - Machine “Cubs” has running Ubuntu 16.04 with python installed, ssh service is on
  - Machine “Indians” is the control panel, wants to install docker on Machine “Cubs” by Ansible

How about after the restored OS boots?

- Only two steps, first define the host file
  - syntax: servername options
    - options:
      - ansible_host -- Remote Host IP
      - ansible_user -- Remote SSH User Name
      - ansible_ssh_private_key_file -- SSH Key
      - ansible_ssh_pass -- SSH Password for remote host
  - E.g.
    - Cubs ansible_host=192.168.11.3 ansible_user=root ansible_ssh_private_key_file=...

How about after the restored OS boots?

- Second, write the playbook file “docker_install.yml” to install docker by Ansible:

```
- name: Install docker and run service
# use group
  hosts: DockerHost
  sudo: True
  tasks:
    - name: Install docker with openSUSE Leap
      zypper: name={{ item }}
      with_items:
        - docker
        - curl
      when: ansible_distribution == "openSUSE Leap"
    - name: Install docker with CentOS
      yum: name={{ item }}
      with_items:
        - docker
```

How about after the restored OS boots?

```
      - curl
    when: ansible_distribution == "CentOS"

  - name: Install docker with Ubuntu
    apt: name={{ item }} update_cache=yes
    with_items:
      - docker.io
      - curl
    when: ansible_distribution == "Ubuntu"
  - name: Create docker link with Ubuntu
    shell: ln -sf /usr/bin/docker.io /usr/local/bin/docker
    when: ansible_distribution == "Ubuntu"
#-------------------------------------------------
  - name: Set docker enable and run
    service: name=docker state=started enabled=yes
```

How about after the restored OS boots?

- On Machine Indians
  - Make sure Ansible is installed, if not
    - http://docs.ansible.com/ansible/intro_installation.html
  - Run the command to install and start docker on machine Cubs:
    - ansible-playbook docker_install.yml

Outline

- Introduction to Clonezilla
  - Features
  - Updates since 2016 Spring
- Unattended system deployment
  - Live boot parameters
  - Ansible
- Big data system deployment
  - Clonezilla-BD
- Q&A

Big data system deployment

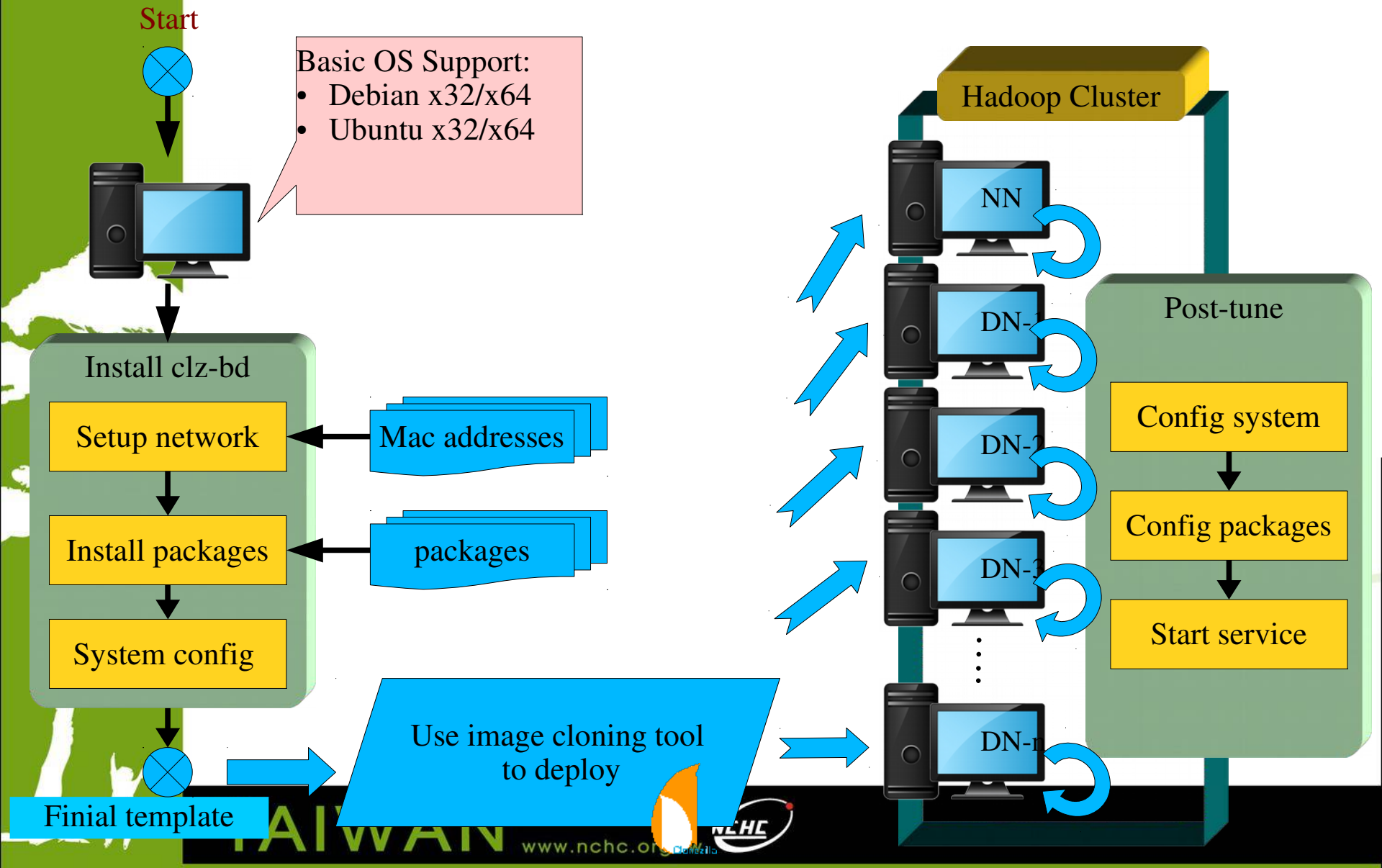
- Massive: many nodes-> need massive deployment
- Complicated: system installation, big data applications

Hadoop Variant		Apache	CDH	HDP	MapR
Description		Hadoop original project	Cloudera Distribution Hadoop	Hortonworks Data Platform (Yahoo subsidiary)	MapR Inc.
License		**Apache license**	**Apache license**	**Apache license**	Proprietary
Deployment from scratch	tools	Apache Ambari	Cloudera Manager	Apache Ambari (customized)	Proprietary
	OS	• RHEL 5/6 • CentOS 5/6 • Oracle Linux 5/6 • SLES 11 • Ubuntu 12.04/14 • Debian 7	• RHEL 5.7/6.4~6.6 • Oracle Linux 5.6/6.4~6.6 • SLES 11 • Ubuntu 12.04/14.04 • Debian 6/7.0/7,1	• RHEL 6x/7x • CentOS 6x/7x • Oracle Linux 6.x/7.x • SLES 11 • Ubuntu 12.04/14.04 • Debian 7x	• RHEL • SLES • Ubuntu
	Notes	• Older GNU/Linux • Apache Ambari 2.1.1.	• Newer GNU/Linux • License fee is required for deploying	• Newer GNU/Linux • Apache Ambari 2.1.1.(customized)	Proprietary

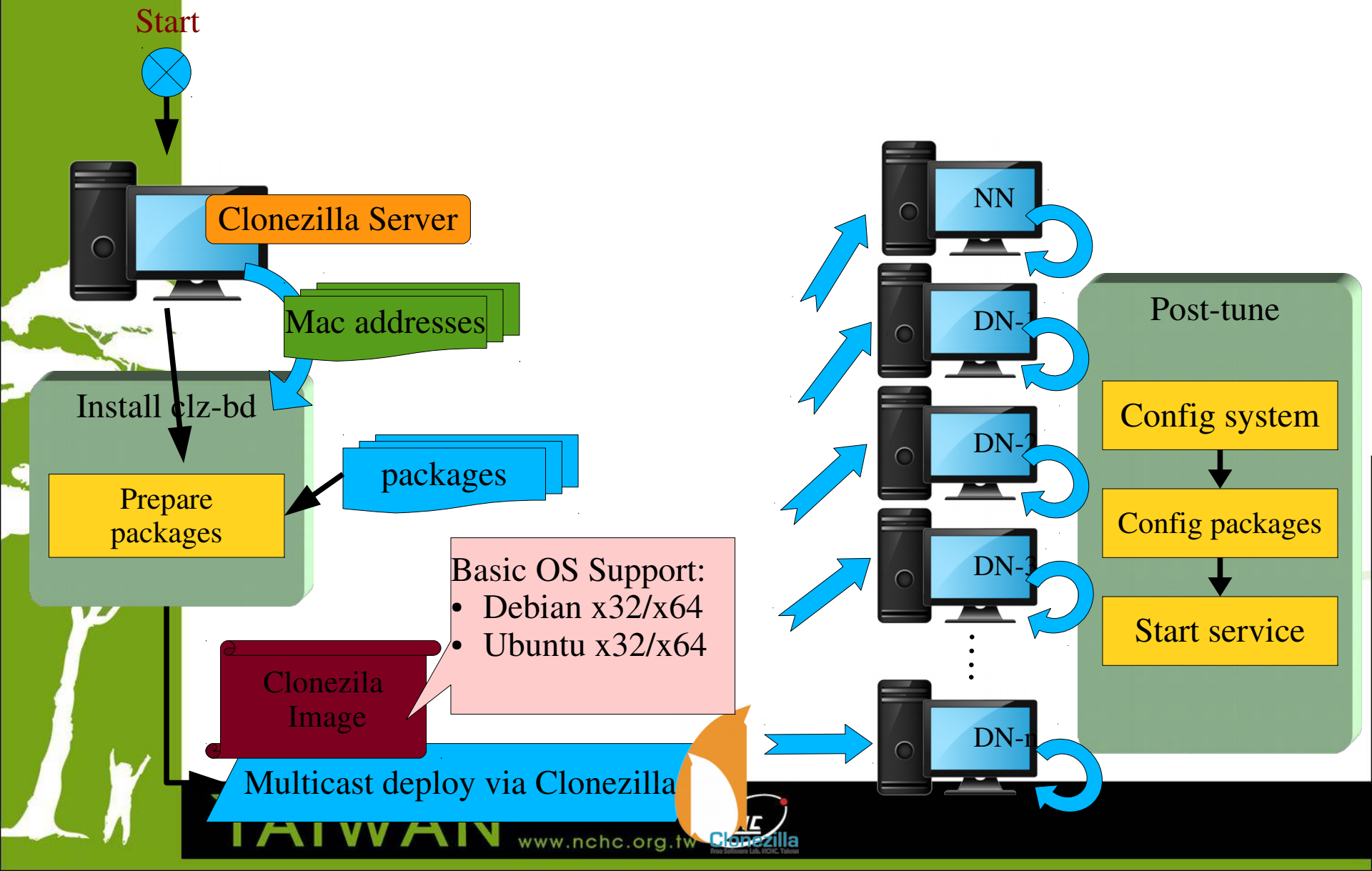
Why Clonezilla-BD?

- A big data deployment program via imaging way
- Easy way to deploy Hadoop cluster
- Pros
  - Easy: auto configuration, including accounts of services, keys, software configuration, services, etc.
  - Compatible**:** works for physical and virtual machine. Can be used with Clonezilla, True Image, etc.
- Two methods
  - Node deployment
  - Clonezilla-SE
- Requirement
  - All MAC addresses for computing nodes should be record in a file for deployment use

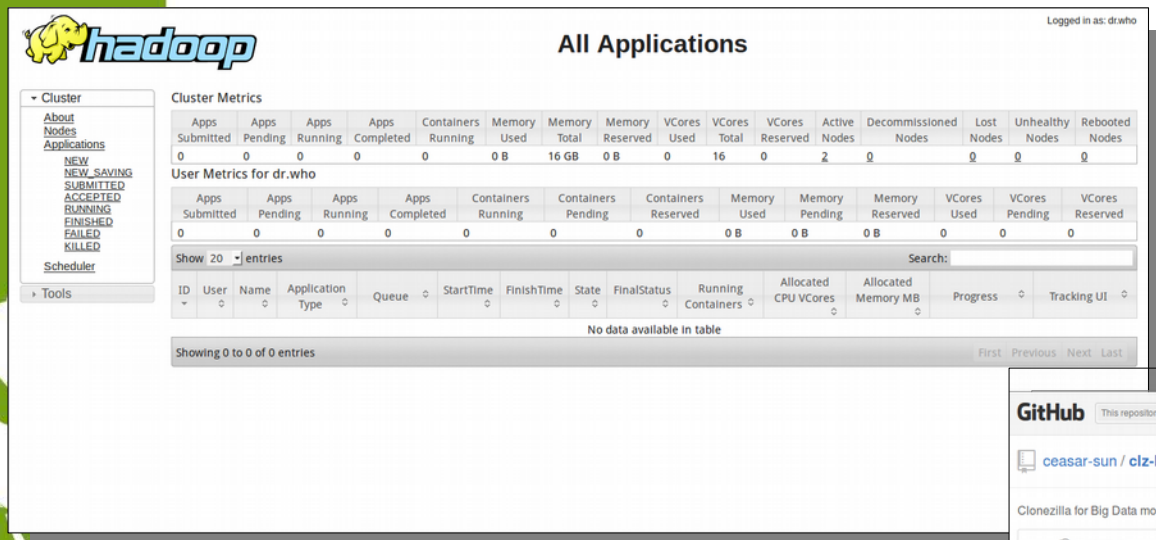
Method 1- node deployment

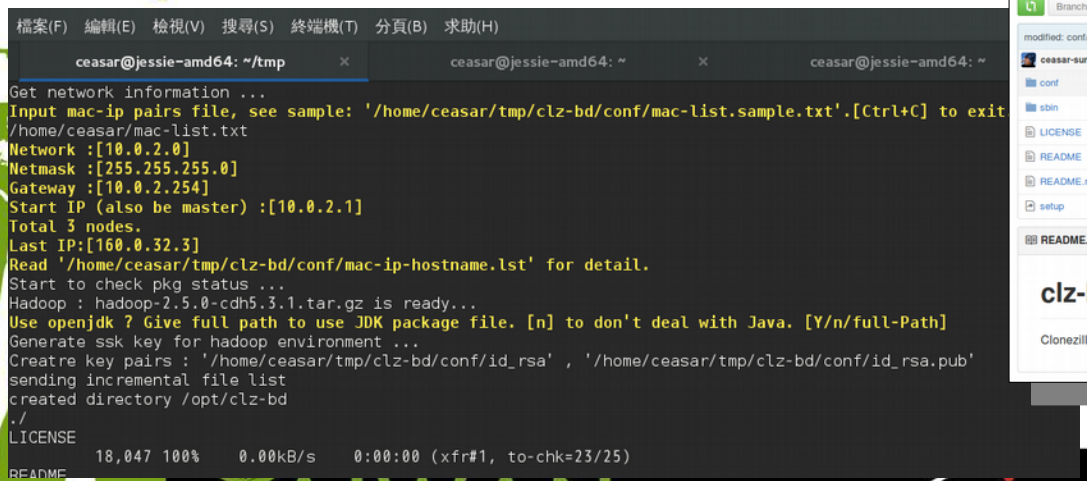


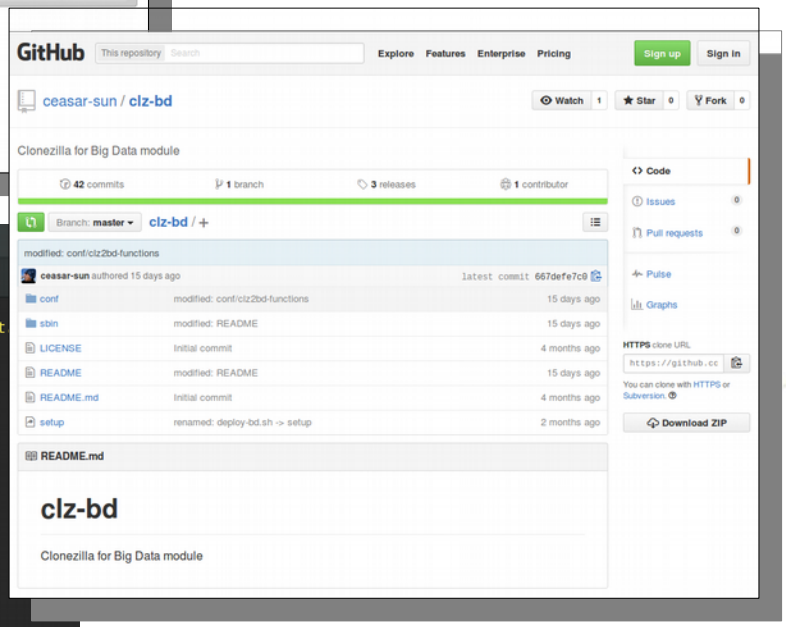
Method 2- Clonezilla-SE



Project and Screenshot







github.com/ceasar-sun/clz-bd

Demo

- Deploy Ubuntu 16.04 (Machine Cubs)
  - Install package python right after Ubuntu 16.04 is restroed.
- Use Ansible to install docker on Machine Cubs, turn on ansible service
- Start a docker environment “busybox”

Conclusion

- With the live boot parameters from Clonezilla live, you can install packages and configure the restored OS right after it's restored.
- With Ansible (or Puppet...), you can install packages and configure the restored OS right after it's rebooted.
- With Clonezilla-BD, you can deploy Hadoop cluster for big data computing.

Reference

- Clonezilla: http://clonezilla.org
- DRBL: http://drbl.org
- Ansible: http://docs.ansible.com/ansible/
- Clonezilla-BD: https://github.com/ceasar-sun/clz-bd

Questions ?

Great!

?????