ZFS file system in desktop lab environment

Dr. Alexei Kotelnikov &
Trent Handlovsky

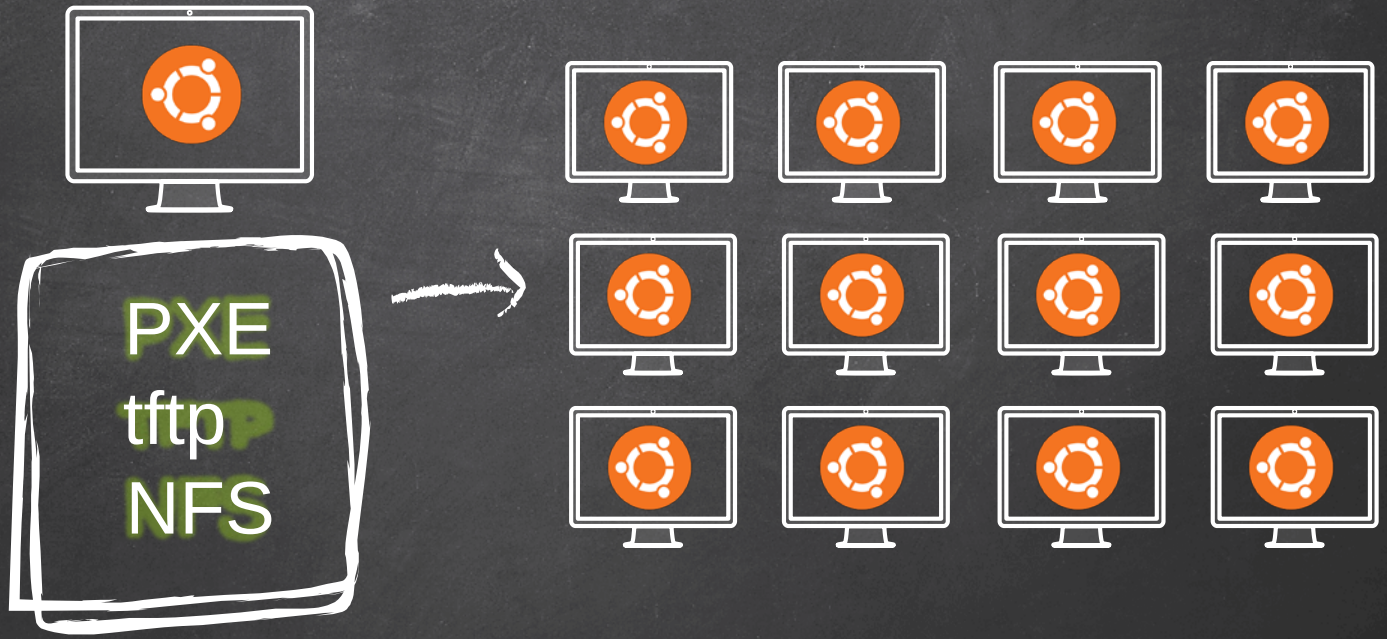
RUTGERS
School of Engineering

Motivation for the Lab Environment

- ✗ Over 5,000 user accounts in Engineering Kerberos
- ✗ Approximately 240 desktops across many labs
  - ✗ Customizable Installation
  - ✗ Quickly Deployable
  - ✗ Easily Upgradeable
- ✗ All previous user data wiped.
- ✗ Need for both Windows and Linux.

Why Linux and ZFS

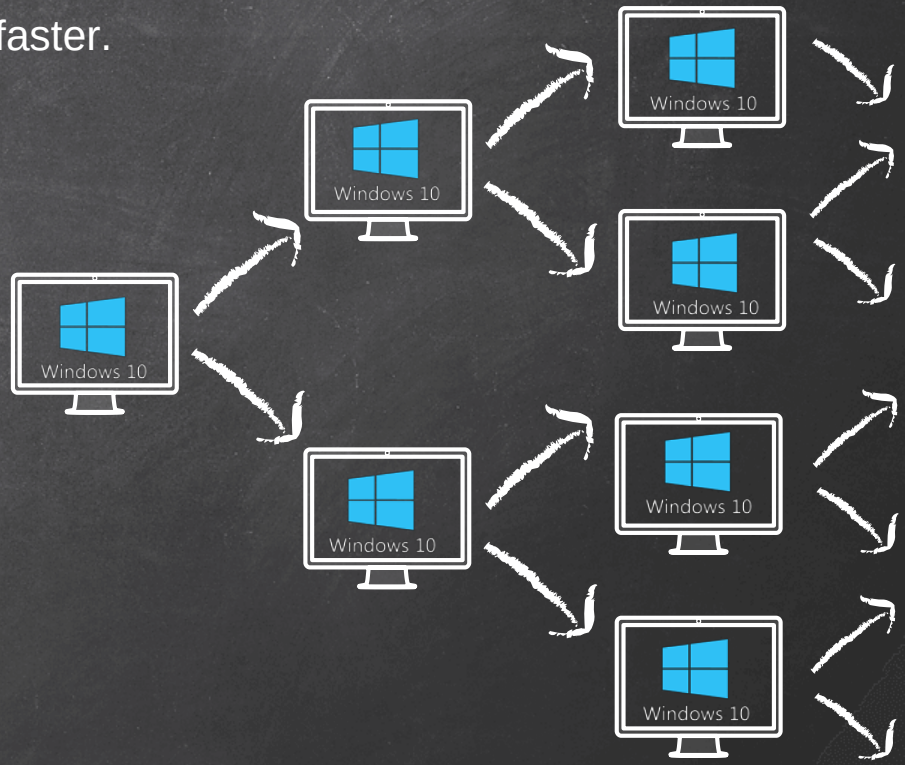
- ✗ Smaller OS install size, shorter install times



Why Linux and ZFS

- ✗ Large Installs are also much faster.

ZFS has the ability to send and receive datasets over the network using a service like SSH or netcat .



What is ZFS?

RUTGERS
School of Engineering

- ZFS is a combined file system and logical volume manager.
- Developed by Sun Microsystems in 2005.
- Now there are two main implementations: by Oracle and by the OpenZFSproject.
- Supported operating systems: Solaris flavors, FreeBSD, NetBSD, Mac OS, and Linux

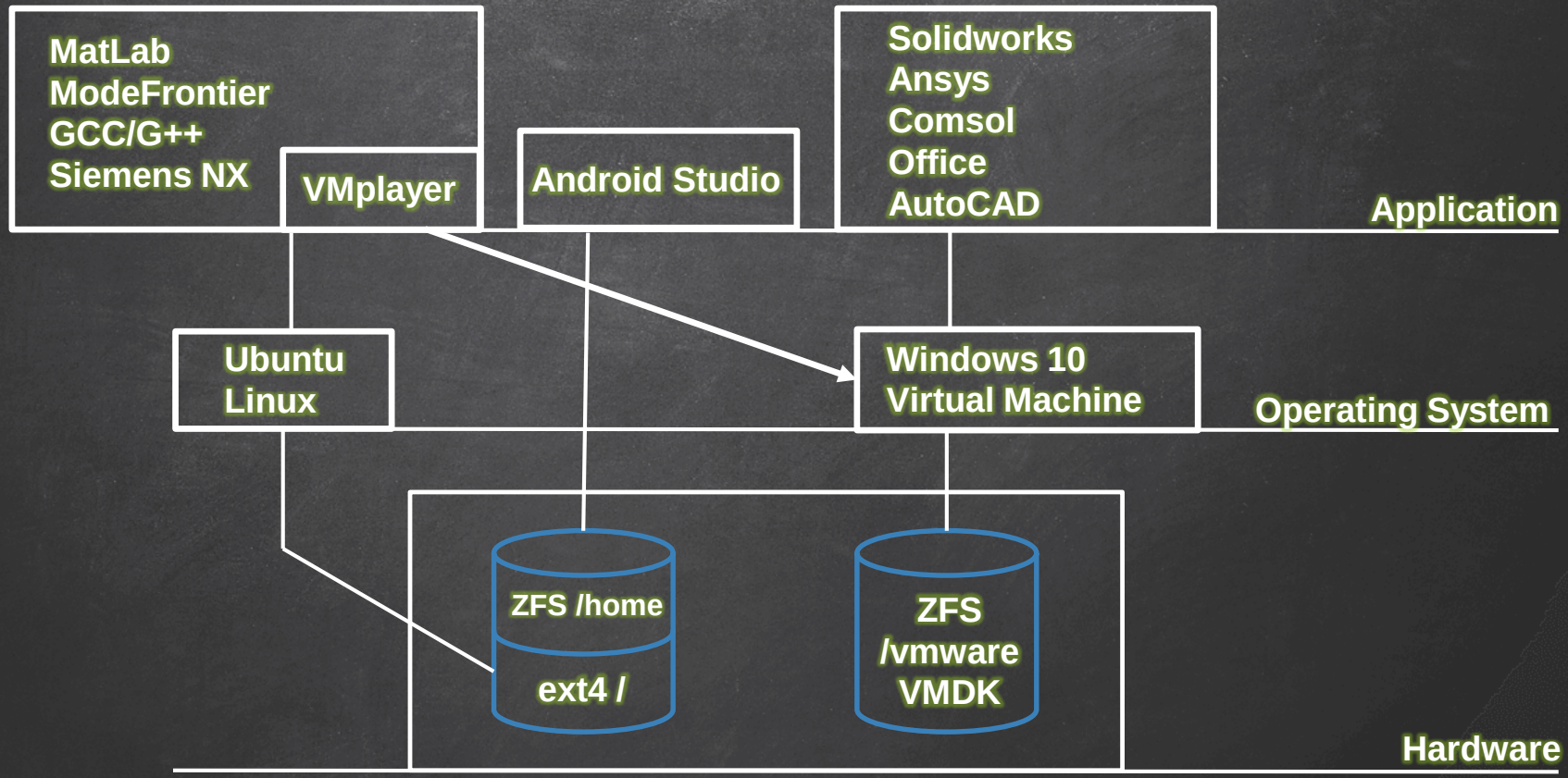
Main features:

- High storage capacity: large volume size (2^128 bytes), file size (2^64 bytes), 2^48 files per directory
- Data management and integrity: snapshots and replication, continuous integrity checking and automatic repair.

Desktop Configuration

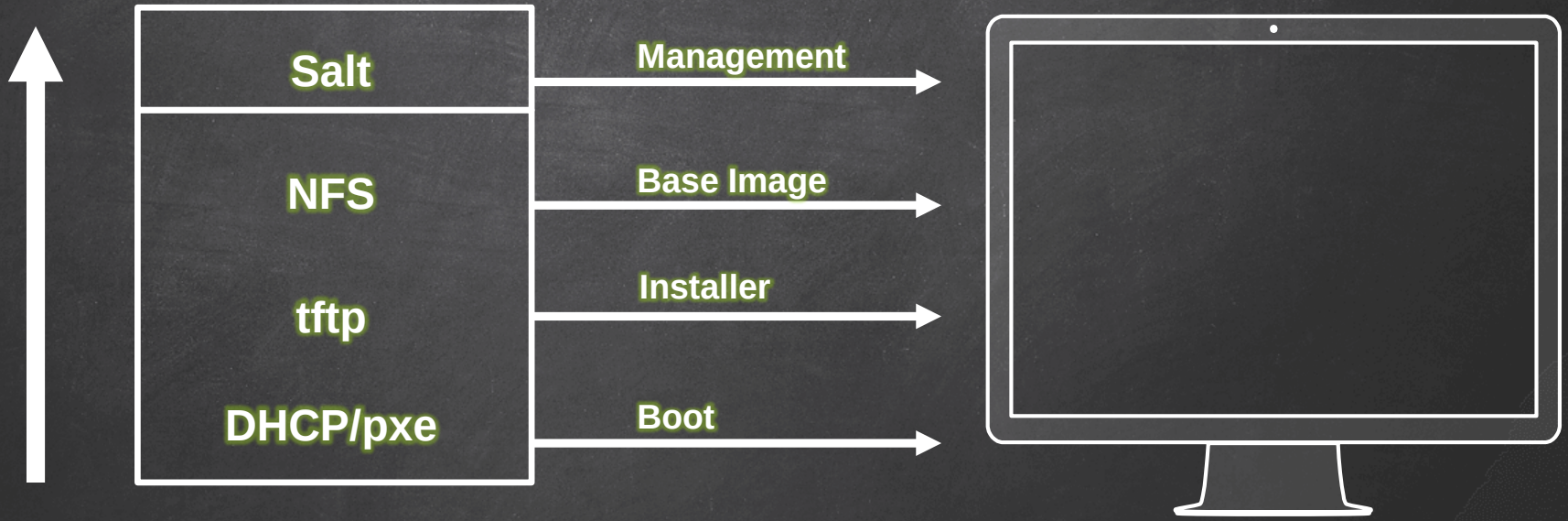
RUTGERS School of Engineering

✗ Smaller OS install size, shorter install times



The Installation

- ✗ Follows an installation/management procedure similar to that of a Linux cluster
- ✗ Total Installation time is about 5 minutes per machine



RUTGERS
School of Engineering

Virtual Machine Deployment via replication


```
root@ELT32:~# zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
home                28.4G  18.8G    96K  none
home/software       13.2G  18.8G  13.2G  /home/software
home/user           15.2G  18.8G  15.1G  /home/user
virtual              162G  68.7G    96K  none
virtual/win7vmware   162G  68.7G   156G  /mnt/vmware
```

```
root@ELT32:~# nc.traditional -w 120 -l -p 8023 | sudo zfs receive $IMAGE -F
```

```
root@ELT32:~# sudo zfs send -R $IMAGE | sudo nc.traditional -w 20 ELT01 8023
```

Virtual Machine Update via incremental

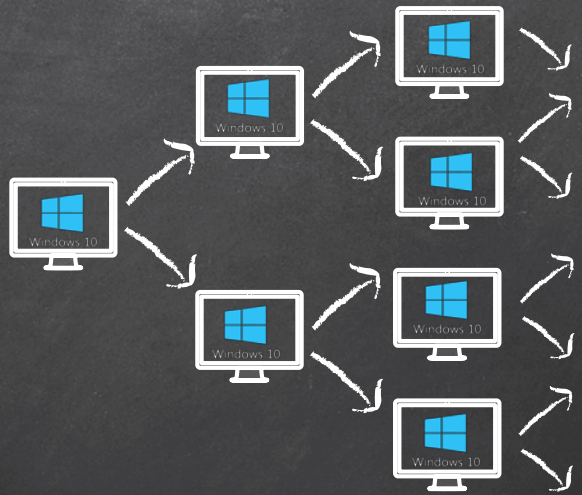

```
root@ELT32:~# zfs list -t snapshot
NAME                                 USED  AVAIL  REFER  MOUNTPOINT
home/software@jan03_18              50.3M      -  13.2G  -
home/user@jan03_18                  48.9M      -  14.9G  -
home/user@jan16_18                  33.3M      -  15.0G  -
virtual/win7vmware@jan11_18         6.02G      -   149G  -
virtual/win7vmware@mar12_18           80K      -   156G  -
```

```
root@ELT32:~# nc.traditional -w 120 -l -p 8023 | sudo zfs receive $IMAGE -F
```

```
root@ELT32:~# zfs send -i $OLD_SNAPSHOT $NEW_SNAPSHOT | nc.traditional ELT01 8023
```

Distributed Deployment

- ✗ Each machine that has a copy of the dataset is able to send it out to multiple machines simultaneously.
- ✗ Machines that have finished receiving the image are then able to send it to others.
- ✗ As the image is being distributed the computers with the image increase in an exponential form.



Sandbox User Environment

RUTGERS School of Engineering

- ZFS uses snapshots to create checkpoints of a filesystem, and has the ability to return, or “rollback”
- Updates can be easily saved by admins and any changes made by users are undone as the filesystem rolls back to the last saved snapshot upon the next login.
- This is accomplished by a script that is executed by a PAM module upon successful authentication with the Kerberos server.

/etc/pam.d/common-session

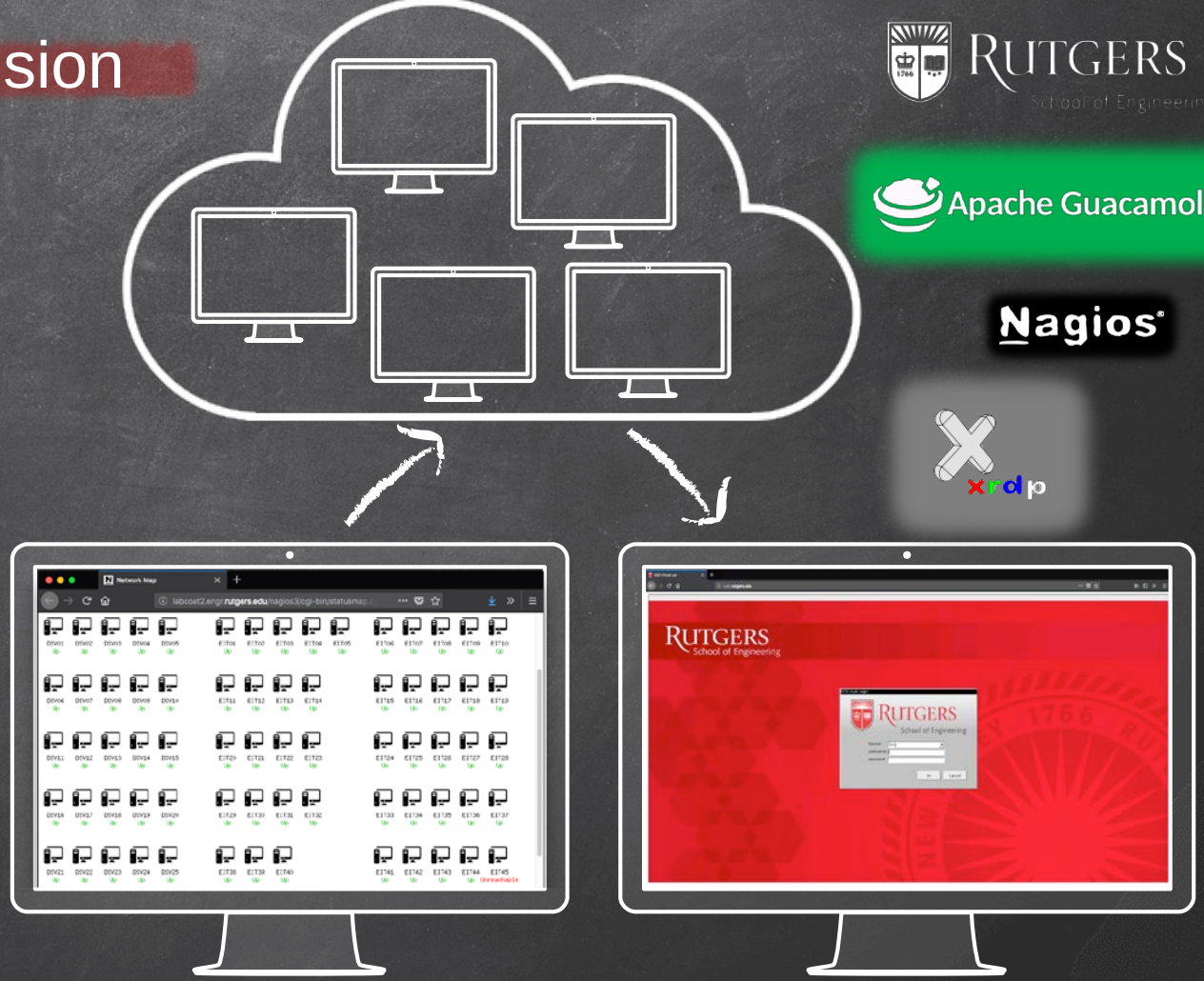
```
# and here are more per-package modules (the "Additional" block)
session optional                        pam_krb5.so minimum_uid=10000
session optional                        pam_script.so
```

/usr/share/libpam-script/pam_script_ses_open

```
#!/bin/bash

env > /tmp/uid

who -u | cut -d' ' -f1 > /tmp/killoldusers

echo `cat /tmp/uid | grep PAM_USER | cut -d'=' -f2` > /tmp/usern
snap_uid=$(id -u `cat /tmp/usern`)

if [ "$snap_uid" -gt 1005 ]
then
/sbin/zfs list -t snapshot | grep home > /tmp/test_home
snap_home=$(tail -1 /tmp/test_home | cut -d' ' -f1)
/sbin/zfs rollback -r $snap_home

/sbin/zfs list -t snapshot | grep virtual > /tmp/test_virtual
snap_virtual=$(tail -1 /tmp/test_virtual | cut -d' ' -f1)
/sbin/zfs rollback -r $snap_virtual

chown -R $PAM_USER:root /home/user
```

Virtual Lab Extension

- ✗ The Virtual Lab allows access to the desktops remotely via a web browser.
- ✗ When the physical computer labs close, access to the desktops in these computer labs becomes available through.
- ✗ ZFS rollback still works when each user logs in.



Summary

- Base install of Linux that supports ZFS
  - Installed through a PXE boots session
  - Desktop to Desktop; 20 GB Linux install ~5mins,
  - Lab Install (60 Desktops); ~4 hours
- Home partition and 2nd drive are formatted as ZFS
  - ZFS filesystem is then sent through ZFS send/receive
    - This propagation can occur at exponential rates
  - Desktop to Desktop; 150 GB Linux install ~15mins,
  - Lab Install (60 Desktops); ~4 hours
- Script is created so that as users log in, the filesystem is returned to it's saved state
- This process safely, securely, and quickly protects users information, and is able to be extended to other use cases.

Thank you