

# Implementing Linux-Enabled Condor in Windows Computer Labs

Joshua Alexander

Chris Franklin

University of Oklahoma



# What is Condor?

- Condor<sup>®</sup> is a program developed by the University of Wisconsin to allow desktop computers to harness idle time to perform computationally intensive operations.

# Why do you need it?

- Condor<sup>®</sup> provides free computing cycles for scientific and research use, which increases supercomputing capacity by acquiring additional computing time on otherwise idle desktop PCs in campus PC labs.

# Condor: Linux vs. Windows

- Condor inside Linux: full featured
- Condor inside Windows®: “clipped”
  - No autocheckpointing
  - No job automigration
  - No remote system calls
  - No Standard Universe

# Lots of PCs in IT Labs

At many institutions, there are lots of PC labs managed by a central IT organizations.

If the head of IT (e.g., CIO) is on board, then all of these PCs can be Condorized.

But, these labs tend to be Windows® labs, not Linux. So you can't take the Windows® desktop experience away from the desktop users, just to get Condor.

So, how can we have Linux Condor **AND** Windows® desktop on the same PC at the same time?

# Solution Attempt #1: VMware

Attempted solution: VMware

- Linux as native host OS
- Condor inside Linux
- VMware inside Linux
- Windows® inside VMware

Tested on ~200 PCs in IT PC labs (Union, library, dorms, Physics Dept)

In production for over a year

# VMware Disadvantages

Attempted solution: VMware

- Linux as native host OS
- Condor inside Linux
- VMware inside Linux
- Windows® inside VMware

## Disadvantages

- VMware costs money! (Less so now than then.)
- Crashy
- VMware performance tuning (straight to disk) was unstable
- Sensitive to hardware heterogeneity
- Painful to manage
- CD/DVD burners and USB drives didn't work in some PCs.

# A Better Solution: coLinux

Cooperative Linux (coLinux)

<http://www.colinux.org/>

- FREE!
- Runs inside native Windows®
- No sensitivity to hardware type
- Better performance
- Easier to customize
- Smaller disk footprint and lower CPU usage in idle
- Minimal management required (~10 hours/month)





# Network Issue

## Networking

- **Bridged**: Each PC has to have a second IP address, so the institution has to have plenty of spare IP addresses available.
- **MAC Addresses**: CoLinux is able to generate a random MAC Address for the computer, but it chooses addresses from a range of addresses, so there is a slight chance to have a collision. We generate CoLinux MACs based on the Windows computer name/number.

# Monitoring Issue

Condor inside Linux monitors keyboard and mouse usage to decide when to suspend a job.

In coLinux, this is tricky.

Working with James Bley at the University of Kansas, we set up a Visual Basic script on the Windows® side to send the keyboard and mouse information to coLinux.

# Current Status

Currently, we have approximately 700 computers running Condor inside coLinux. These computers have been running without significant problem for several months.

# Future Goals

- Make the installation even easier
- Allow for additional monitoring of keyboard and mouse usage or whether or not a user is logged in

# Questions?



Unclipped Condor in Windows via coLinux

# Implementation

- [http://www.oscer.ou.edu/CondorInstall/condor\\_colinux\\_howto.php](http://www.oscer.ou.edu/CondorInstall/condor_colinux_howto.php)
- 6-Step Process
  - Install WinPcap ([www.winpcap.org](http://www.winpcap.org))
  - Download our 7-zip file and extract it
  - Customize condor\_config
  - Customize first-run linux file first.sh
  - Customize the Windows install script
  - Run it

# Implementation – condor\_config

- We've included our condor\_config file in the distribution file, you'll want to edit at least the following values:
  - CONDOR\_HOST
  - CONDOR\_ADMIN
  - COLLECTOR\_NAME
  - HOSTALLOW\_READ
  - HOSTALLOW\_WRITE

# Implementation – first.sh

- The distribution includes a file named “first.sh”. This file is run inside Linux the first time you boot your coLinux install.
  - “#Create CIFS Mounts” – We mount several CIFS shares. If you wish to mount your own shares, you can edit these lines, otherwise you can just delete them.
  - “#Run OU Update script hourly” – We have a shell script on one share that we run hourly, which allows us to easily apply various updates to the machine. You can edit/remove this section.
  - “#Modify hosts file” – Our site has some DNS problems, so we add entries to the hosts file and run a script to dynamically set the hostname.





# Implementation – Windows Install Script

- Disk/Memory usage can be customized through the Windows Install Script we use.
  - Change the first line, conmem=512. We allow each machine to use up to half of its memory for coLinux
  - Change the values on the FSUtil lines to change disk usage. We allocate 16 Gig for Condor local files, and 2 gig for swap. These values are far higher than necessary, and can be significantly reduced.
  - Part of our script assigns coLinux unique MAC addresses based on the computer name. You can either modify this to fit your computer names, or allow coLinux to randomly assign MAC addresses.
- Boot.ini – We copy in a new boot.ini file with the necessary switches, if your boot.ini file isn't compatible, you'll need to edit the file

# Questions?



Unclipped Condor in Windows via coLinux