

# Implementing Linux-Enabled Condor in Windows Computer Labs

Horst Severini,  
Chris Franklin, Josh Alexander, Joel  
Snow

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<sup>®</sup>: “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<sup>®</sup> labs, not Linux. So you can't take the Windows<sup>®</sup> desktop experience away from the desktop users, just to get Condor.

So, how can we have Linux Condor **AND** Windows<sup>®</sup> 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)



Unclipped Condor in Windows via coLinux  
HiPCAT 2008



# Preventing BSOD

- The Data Execution Prevention feature inside Windows<sup>®</sup>, when running on some newer processors, can conflict with coLinux and cause system failure. The solution to this problem is to add the /NOEXECUTE switch to the Windows<sup>®</sup> boot.ini.

# Network Issue

## Networking options

- **Bridged**: Each PC has to have a second IP address, so the institution has to have plenty of spare IP addresses available. (Oklahoma solution)
- **NAT**: The Condor pool requires a Generic Connection Broker (GCB) on a separate, dedicated PC (hardware \$), and has some instability. Switched to OpenVPN.(Nebraska solution)
  - Nebraska experimented with port forwarding in Windows<sup>®</sup>, but abandoned it for OpenVPN because of security and usability.



# 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<sup>®</sup> 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.

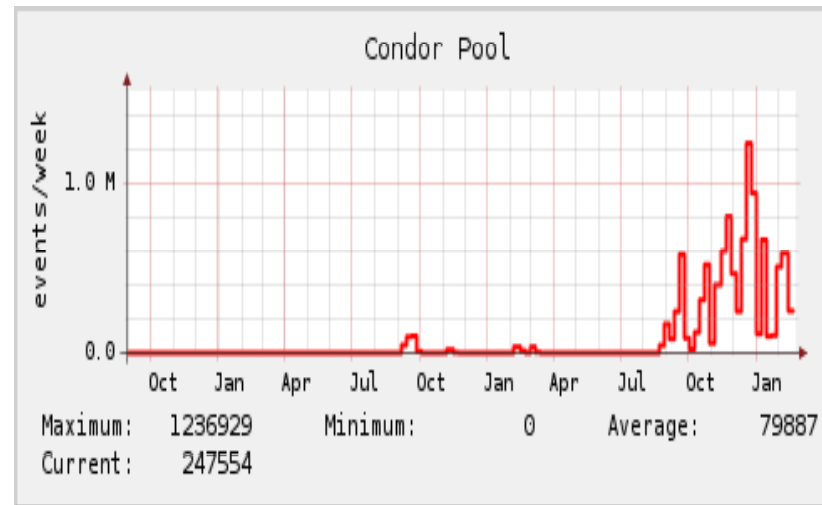
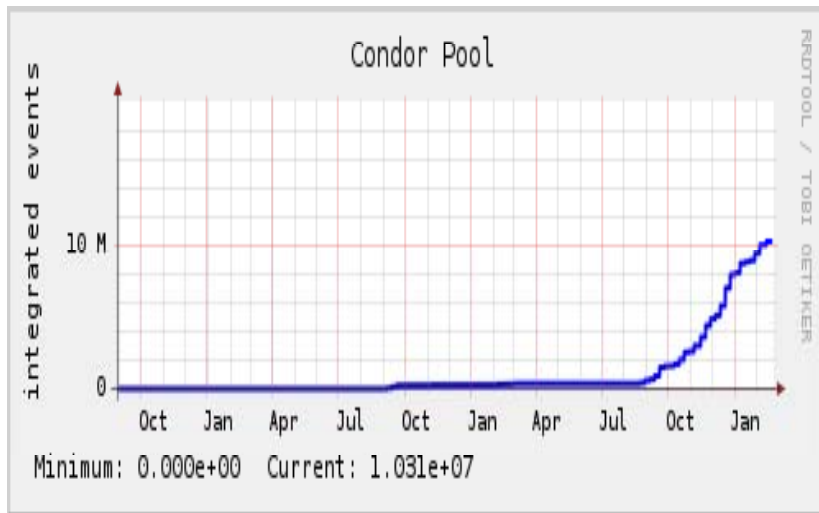
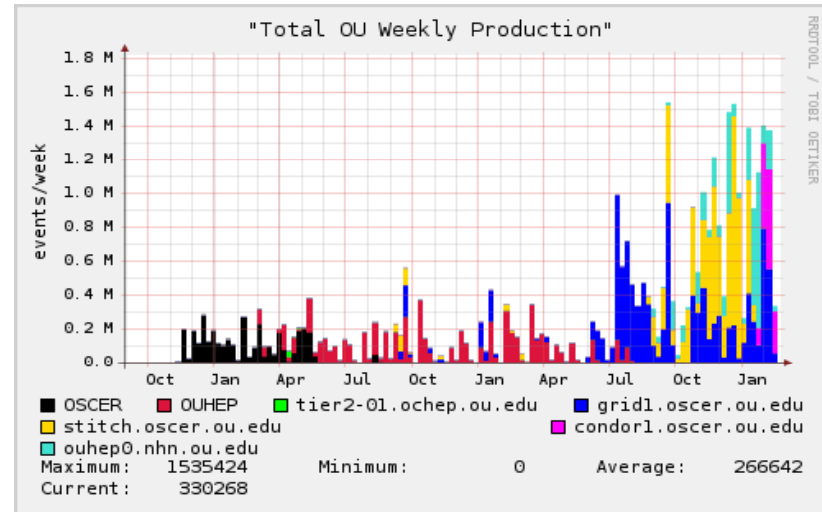
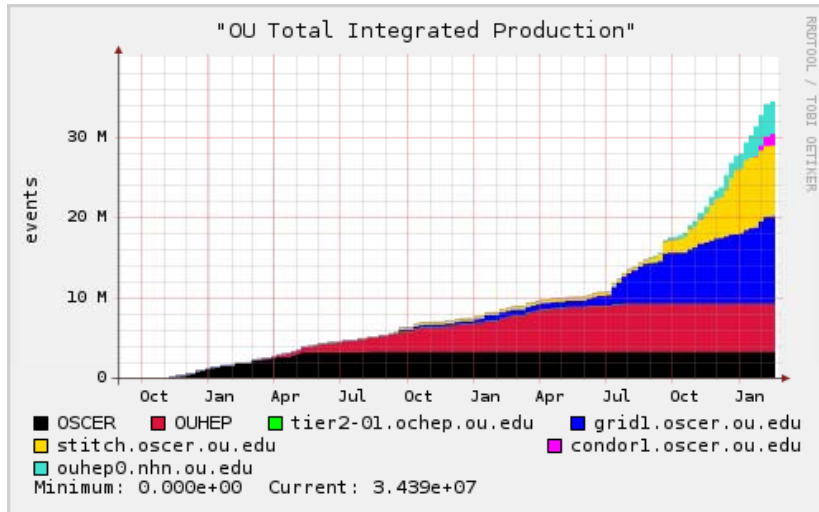


# Future Goals

- Make the installation even easier
- Allow for additional monitoring of keyboard and mouse usage
- Vista compatibility



# MC Production at OU



Unclipped Condor in Windows via coLinux  
HiPCAT 2008

# Questions?



Unclipped Condor in Windows via coLinux  
HiPCAT 2008