Implementing Linux-Enabled Condor in Windows Computer Labs

Horst Severini

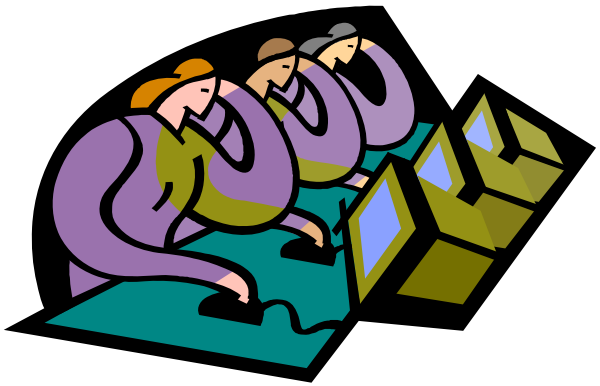
Chris Franklin, Josh Alexander

University of Oklahoma

Opportunistic Computing

What is Opportunistic Computing?

Desktop PCs Are Idle Half the Day



Desktop PCs tend to be active during the workday.

But at night, during most of the year, they're idle. So we're only getting **half their value** (or less).

Supercomputing at Night

A particular institution – say, OU – has lots of desktop PCs that are **idle during the evening and during intersessions**.

Wouldn't it be great to put them to work on something **useful** to our institution?

That is: What if they could pretend to be a big supercomputer **at night**, when they'd **otherwise be idle anyway**?

This is sometimes known as ***opportunistic computing***: When a desktop PC is otherwise idle, you have an opportunity to do number crunching on it.

Supercomputing at Night Example

SETI – the Search for Extra-Terrestrial Intelligence – is looking for evidence of green bug-eyed monsters on other planets, by mining radio telescope data.

SETI@home runs number crunching software as a screensaver on idle PCs around the world (1.6 million PCs in 231 countries):

http://setiathome.berkeley.edu/

There are **many similar projects**:

- folding@home (protein folding)
- climateprediction.net
- Einstein@Home (Laser Interferometer Gravitational wave Observatory)
- Cosmology@home
- …

BOINC

The projects listed on the previous page use a software package named BOINC (**B**erkeley **O**pen **I**nfrastructure for **N**etwork **C**omputing), developed at the University of California, Berkeley:

http://boinc.berkeley.edu/

To use BOINC, you have to insert calls to various BOINC routines into your code. It looks a bit similar to MPI:

```
int main ()
{ /* main */
    …
    boinc_init();
    …
    boinc_finish(…);
} /* main */
```

Condor is Like BOINC

- Condor **steals computing time** on existing desktop PCs **when they're idle**.
- Condor **runs in background** when no one is sitting at the desk.
- Condor allows an institution to get **much more value** out of the hardware that's **already purchased**, because there's little or no idle time on that hardware – all of the idle time is used for number crunching.

Condor is Different from BOINC

- To use Condor, **you don't need to rewrite your software** to add calls to special routines; in BOINC, you do.
- Condor **works great under Unix/Linux**, but less well under Windows or MacOS (more on this presently); BOINC works well under all of them.
- It's **non-trivial to install Condor** on your own personal desktop PC; it's straightforward to install a BOINC application such as SETI@home.

Useful Features of Condor

- **Opportunistic** computing: Condor steals time on existing desktop PCs when they're otherwise not in use.
- Condor **doesn't require any changes to the software**.
- Condor can **automatically checkpoint** a running job: every so often, Condor saves to disk the state of the job (the values of all the job's variables, plus where the job is in the program).
- Therefore, Condor can **preempt** running jobs if more important jobs come along, or if someone sits down at the desktop PC.
- Likewise, Condor can **migrate** running jobs to other PCs, if someone sits at the PC or if the PC crashes.
- And, Condor can do all of its **I/O over the network**, so that the job on the desktop PC doesn't consume the desktop PCs local disk.

Condor Limitations

- The Unix/Linux version has **more features** than Windows or MacOS, which are referred to as “clipped.”
- Your code **shouldn’t be parallel** to do opportunistic computing (MPI requires a fixed set of resources throughout the entire run), and it shouldn’t try to do any funky communication (e.g., opening sockets).
- For a Red Hat Linux Condor pool, you have to be able to **compile your code** with gcc, g++, g77 or NAG f95.
- Also, depending on the PCs that have Condor on them, you may have limitations on, for example, how big your jobs’ RAM footprint can be.

Why do you need it?

- Condor® 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.

Running a Condor Job

Running a job on Condor pool is a lot like running a job on a cluster:

1. You compile your code using the compilers appropriate for that resource.
2. You submit a batch script to the Condor system, which decides when and where your job runs, magically and invisibly.

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

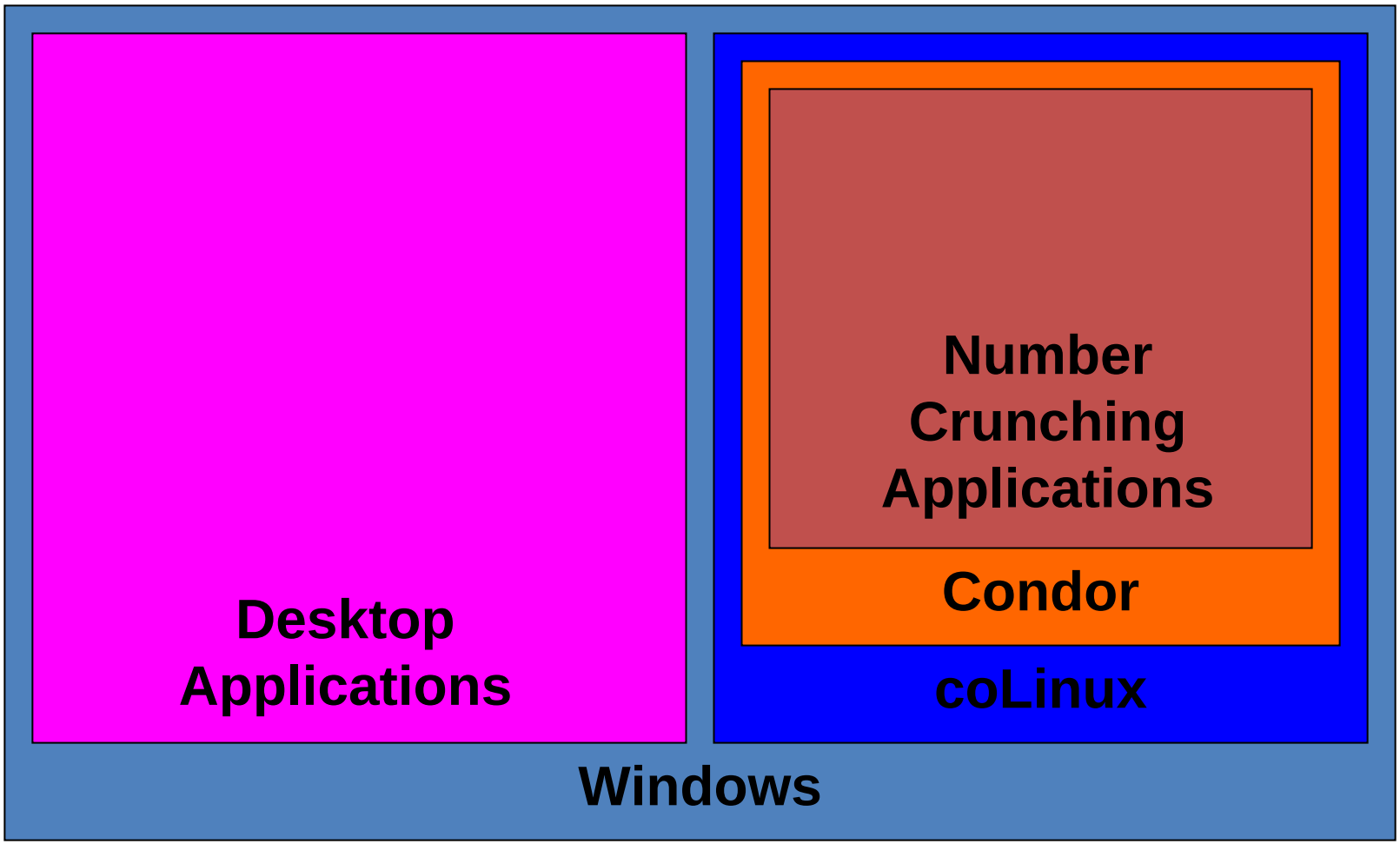
- VMware costs money! (Less so now than then.)
- Crashy
- VMware performance tuning (straight to disk) was unstable
- Sensitive to hardware heterogeneity
- Painful to manage
- CD/DVD burners and USB drives didn't work in some PCs.

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)

Condor inside Linux inside Windows



Advantages of Linux inside Windows

- Condor is full featured rather than clipped.
- Desktop users have a full Windows experience, without even being aware that coLinux exists.
- A little kludge helps Condor watch the keyboard, mouse and CPU level of Windows, so that Condor jobs don't run when the PC is otherwise in use.

Want to try it yourself?

http://www.oscer.ou.edu/CondorInstall/condor_colinux_howto.php

Network Issues

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®, but abandoned it for OpenVPN because of security and usability.

Monitoring Issues

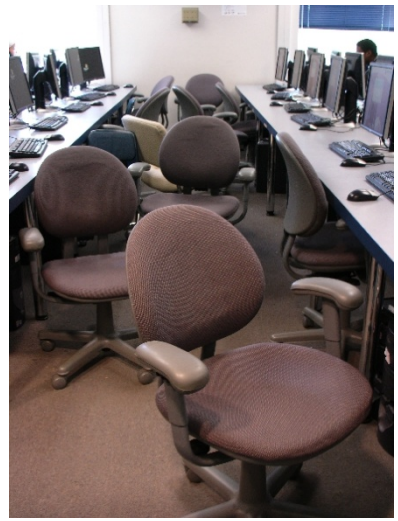
Condor inside Linux monitors keyboard and mouse usage to decide when to suspend a job.

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.

Our Condor Pool

- Two Head Nodes
  - Condor1
  - Condor2
  - Each runs condor_schedd
- One Condor pool
  - "Default" pool across campus
    - 775 desktop PCs in dozens of labs around campus
  - Each computer runs a startd



Our Condor Pool

- Unfortunately only 325 machines appear in the pool.
- Reasons:
  - Recent hardware and software upgrades in computer labs
  - Some machines were recently moved to a new location and have not been put back into service.
  - Unknown network problems in one lab

Current Status of Project

Partnering with other institutions

- Oklahoma State University
- University of Southern Alabama
- University of Texas – Arlington

Other Institutions Interested

- Costa Rica
- University of South Dakota
- Tanzania

Current Status of Project

- Software and installation instructions available for download

  http://www.oscer.ou.edu/CondorInstall/condor_colinux_howto.php

Future Goals

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

OU's NSF CI-TEAM Project

OU's NSF CI-TEAM Project

OU recently received a grant from the National Science Foundation's Cyberinfrastructure Training, Education, Advancement, and Mentoring for Our 21st Century Workforce (CI-TEAM) program.

Objectives:

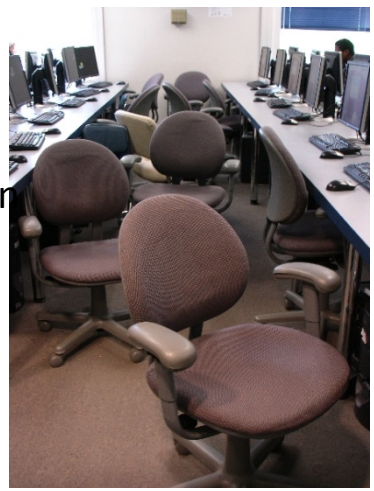
- Provide Condor resources to the national community
- Teach users to use Condor and sysadmins to deploy and administer it
- Teach bioinformatics students to use BLAST over Condor

OU NSF CI-TEAM Project

Cyberinfrastructure Education for Bioinformatics and Beyond

Objectives:

- **teach** students and faculty to use **FREE** Condor middleware, stealing computing time on idle PCs;
- **teach** system administrators to deploy and maintain Condor on PCs;
- **teach** bioinformatics students to use BLAST or Condor;
- **provide** Condor Cyberinfrastructure to the national community (**FREE**).



OU will provide:

- **Condor pool** of 775 desktop PCs (already part of the **Open Science Grid**);
- ***Supercomputing in Plain English*** workshops via videoconferencing;
- Cyberinfrastructure ***rounds*** (consulting) via videoconferencing;
- **Instructions** for installing full-featured Condor on a Windows PC (Cyberinfrastructure for **FREE**);
- **sysadmin consulting** for installing and maintaining Condor on desktop PCs.

OU's team includes: High School, Minority Serving, 2-year, 4-year, masters-granting; 18 of the 32 institutions are in 8 EPSCoR states (AR, DE, KS, ND, NE, NM, OK, WV).

OU NSF CI-TEAM Project

Participants at OU

(29 faculty/staff in 16 depts)

- Information Technology
  - OSCER: **Neeman (PI)**
- College of Arts & Sciences
  - Botany & Microbiology: Conway, Wren
  - Chemistry & Biochemistry: **Roe (Co-PI)**, Wheeler
  - Mathematics: White
  - Physics & Astronomy: Kao, **Severini (Co-PI)**, Skubic, Strauss
  - Zoology: Ray
- College of Earth & Energy
  - Sarkeys Energy Center: Chesnokov
- College of Engineering
  - Aerospace & Mechanical Engr: Striz
  - Chemical, Biological & Materials Engr: Papavassiliou
  - Civil Engr & Environmental Science: Vieux
  - Computer Science: Dhall, Fagg, Hougen, Lakshmivarahan, McGovern, Radhakrishnan
  - Electrical & Computer Engr: Cruz, Todd, Yeary, Yu
  - Industrial Engr: Trafalis
- OU Health Sciences Center, Oklahoma City
  - Biochemistry & Molecular Biology: Zlotnick
  - Radiological Sciences: **Wu (Co-PI)**
  - Surgery: Gusev

Participants at other institutions

(62 faculty/staff at 31 institutions in 18 states)

1. California State U Pomona (masters-granting, minority serving): Lee
2. Colorado State U: Kalkhan
3. Contra Costa College (CA, 2-year, minority serving): Murphy
4. Delaware State U (masters, EPSCoR): Lin, Mulik, Multnovic, Pokrajac, Rasamny
5. Earlham College (IN, bachelors): Peck
6. East Central U (OK, masters, EPSCoR): Crittell,Ferdinand, Myers, Walker, Weirick, Williams
7. Emporia State U (KS, masters-granting, EPSCoR): Ballester, Pheatt
8. Harvard U (MA): King
9. Kansas State U (EPSCoR): Andresen, Monaco
10. Langston U (OK, masters, minority serving, EPSCoR): Snow, Tadesse
11. Longwood U (VA, masters): Talaiver
12. Marshall U (WV, masters, EPSCoR): Richards
13. Navajo Technical College (NM, 2-year, tribal, EPSCoR): Ribble
14. Oklahoma Baptist U (bachelors, EPSCoR): Chen, Jett, Jordan
15. Oklahoma Medical Research Foundation (EPSCoR): Wren
16. Oklahoma School of Science & Mathematics (high school, EPSCoR): Samadzadeh
17. Purdue U (IN): Chaubey
18. Riverside Community College (CA, 2-year): Smith
19. St. Cloud State University (MN, masters): J. Herath, S. Herath, Guster
20. St. Gregory's U (OK, 4-year, EPSCoR): Meyer
21. Southwestern Oklahoma State U (masters, EPSCoR, tribal): Linder, Moseley, Pereira
22. Syracuse U (NY): Stanton
23. Texas A&M U-Corpus Christi (masters): Scherger
24. U Arkansas Fayetteville (EPSCoR): Apon
25. U Arkansas Little Rock (masters, EPSCoR): Hall, Jennings, Ramaswamy
26. U Central Oklahoma (masters-granting, EPSCoR): Lemley, Wilson
27. U Illinois Urbana-Champaign: Wang
28. U Kansas (EPSCoR): Bishop, Cheung, Harris, Ryan
29. U Nebraska-Lincoln (EPSCoR): Swanson
30. U North Dakota (EPSCoR): Bergstrom, Hoffman, Majidi, Moreno, Peterson, Simmons, Wiggen, Zhou
31. U Northern Iowa (masters-granting): Gray

Are you interested?

As part of the CI-TEAM, NSF grant I will help you establish your very own condor pool.

- Contact us at:
- jalexander@ou.edu
- hs@nhn.ou.edu
- hneeman@ou.edu
- chrisfranklin@ou.edu

Questions?