

Introduction to Condor

(and using it at Lehigh University)

Ashutosh Mahajan

Department of Industrial and Systems Engineering
Lehigh University

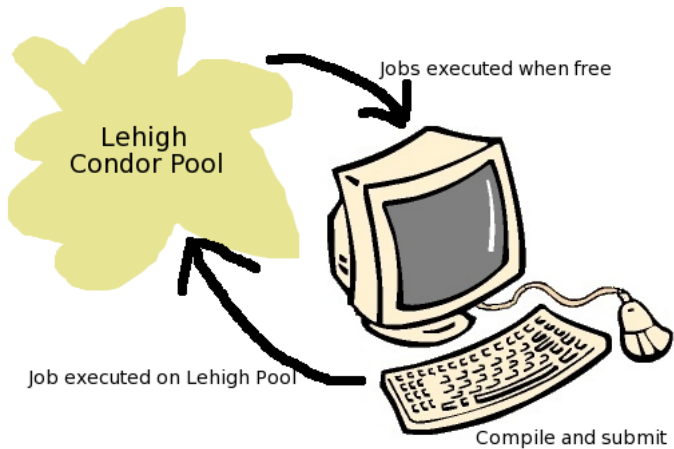
HPC day, Lehigh, April 21, 2006

What is Condor

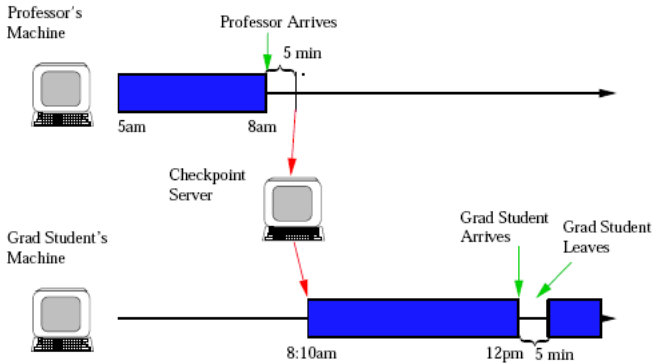
- How do you run a small program?
- What do you do if you want to run a **BIG** program?
- What do you do if
 - you want to run a **BIG** program?
 - you want to run it many times?
 - several people want to do this at the same time?
 - there are only a limited number of machines available.

Start Pooling resources!

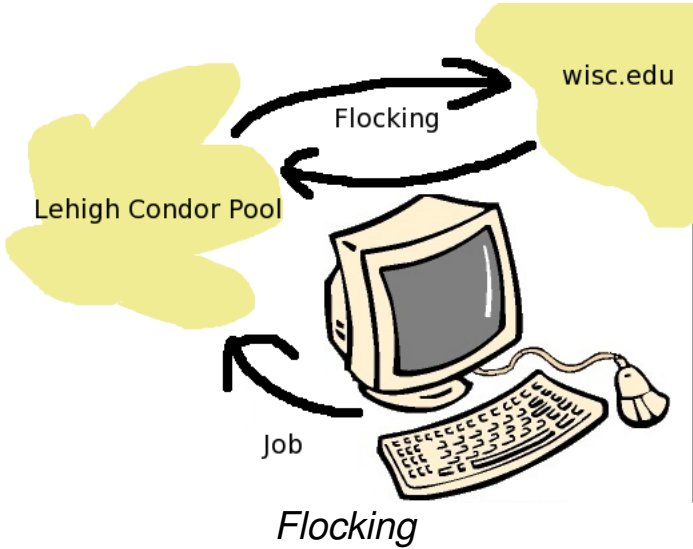
Condor is a **resource scheduler** for computing systems.



Pooling resources



Checkpointing and migration





Condor

High Throughput Computing

*Condor is a **lot** more than a scheduler*

- Can use non-dedicated resources. [Use a machine only when free]
- Checkpoint and migrate
- Remote I/O
- Flocking to other pools

Fire and forget

Users submit their serial or parallel jobs to Condor, Condor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion.



What can condor run

Anything!

- Simple user compiled programs: c, c++, fortran, scripts ...
- Parallel programs
- System commands
- Compilation/installation of condor!

Condor multiverse

- Vanilla: Simple programs
- Standard: Programs with checkpointing: Needs linking with condor libraries.
- Parallel: Parallel programs. MPI/PVM/anything else!. Needs dedicated machines.
- Java: Out of scope in this presentation
- Many more.

Condor at Lehigh

- Blaze: 128 processors, 64-bit
- Fire: 96
- Cor@l: 45
- Vega: 621

Starting with condor

Log in

```
ssh hpcxxx@blaze1.cc.lehigh.edu
```

Setting up

```
[hpc40@blaze1 ]$ scp -r /home/asm4/condor .  
[hpc40@blaze1 ]$ cd condor  
[hpc40@blaze1 condor]$ condor -version
```

Setting up

```
[hpc40@blaze1 condor]$ echo $PATH  
$ export PATH=/usr/local/condor/bin:::$PATH
```

Looking at condor

Status

```
[hpc40@blazel1 condor]$ condor_status
[hpc40@blazel1 condor]$ condor_status -total
[hpc40@blazel1 condor]$ condor_status -server
[hpc40@blazel1 condor]$ condor_status -server
[hpc40@blazel1 condor]$ condor_status -submitters
[hpc40@blazel1 condor]$ condor_q asm4 -analyze
[...]$ condor_status -constraint 'RemoteUser ==
"bad0.cc.lehigh.edu"'
```

Looking at condor

Queues and Priorities

```
[hpc40@blaze1 condor]$ condor_q asm4  
[hpc40@blaze1 condor]$ condor_q asm4 -long : less  
[hpc40@blaze1 condor]$ condor_userprio -allusers  
[hpc40...]$ condor_userprio -all -allusers
```

Running simple jobs

Submit in a vanilla universe

```
[hpc40@blaze1 condor]$ cd serial
[hpc40@blaze1 serial]$ make
[hpc40@blaze1 serial]$ cat partition.condor
$ condor_submit partition.condor
$ condor_status -submitters
$ condor_rm hpcxxx
$ condor_submit partition.condor
$ condor_status -submitters
$ condor_rm hpcxxx
```

Running simple jobs

Submit in a standard universe

- 1 needs to be linked with condor libraries
- 2 condor_compile
- 3 rest is same as a vanilla universe

Running MPI jobs

Submit in a parallel universe

```
[hpc40@blaze1 condor]$ cd mpi
[hpc40@blaze1 condor]$ make
[hpc40@blaze1 condor]$ cat pi.condor
[hpc40@blaze1 condor]$ condor_submit pi.condor
$ condor_status -submitters
```

- ssh lehigh-id@vega.cc.lehigh.edu and see condor there.
- <http://www.cs.wisc.edu/condor/> for more information.