



**There are many clouds like it,
but this one is mine.
(condor_annex)**

condor_annex?

- The (Amazon-only) prototype of a tool to make it simpler to expand an HTCCondor pool to the cloud.
- Will discuss motivations & plans rather than technical details that will change before you get a chance to use it.

motivating example

- Dr. Needs-Moore needs more cycles in less time than her local pool can provide.
- Let's suppose she needs to run 10000 one-hour jobs (in addition to whatever her local pool will provide) by the end of the week.
- She decides she's willing to spend some money to make this happen.

without condor_annex

- › Decide which type(s) of instances to use.
- › Pick a machine image, install HTCondor.
- › Configure HTCondor:
 - to join the pool. (Coordinate with pool admin.)
 - to shut down instance when not running a job (because of the long tail or a problem somewhere)
- › Decide on a bid for each instance type, according to its location (or pay more).
- › Configure the network and firewall at Amazon.
- › Implement a fail-safe in the form of a lease to make sure the pool does eventually shut itself off.
- › Automate response to being out-bid.

with condor_annex

- › Simplified to a single command:

```
condor_annex --annex-id 'TheNeeds-MooreLab' \  
  --expiry '2015-12-18 23:59' \  
  --instances 1
```

- › This usage assumes that the pool administrator has configured defaults.
- › Only one instance at first to test.

process with condor_annex

- Dr. Needs-More submits (one of) her ten thousand jobs to make sure it works.
- When it succeeds, she can make the annex larger just as easily:

```
condor_annex --annex-id 'TheNeeds-MooreLab' \  
  --instances 100
```

[Demo]

condor_annex (complicated)

```
condor_annex --annex-id=TheNeeds-MooreLab \  
  --expiry="2016-04-06 17:00" \  
  --instances=16 \  
  --keypair=Needs-Moore\  
  --vpc=vpc-abcdef12 \  
  --subnet=subnet-123..., subnet-234..., subnet-345... \  
  --image-ids=ami-91e1a3fb \  
  --spot-prices=0.06 \  
  --instance-types=m3.medium \  
  --central-manager=cm.example.wisc.edu \  
  --password-file=../password_file \  
  --region=us-west-2
```

good intentions

- › Manage cloud account credentials.
- › Add explicit budgets.
 - `--dollar-limit 100?`
- › Automate cost optimization.
 - `--request-memory 1024?`
 - `--request-cpu 1 ?`

more good intentions

- Work with more cloud providers.
- Monitor annex instances.
 - New daemon will poll the cloud for the status of each annex and send an aggregate ad to the collector. (`condor_status -annex`)
- Allow keep-alives. (Very short leases regularly renewed by HTCondor daemon.)

HTCondor API?

- **Expose** `condor_annex` mechanisms:
 - Efficient large-scale instance provisioning.
 - Leases for cloud resources.
 - Cost optimization.
 - Automated secure distribution of credentials.

conclusion

- Annex concept has been proven.
- Production code in the pipeline.
- Final design will, as always, be strongly influenced by our ongoing collaboration with you.

[Demo]

questions?

- › contact me at tlmiller@cs.wisc.edu