Upgrading Condor

Best Practices

Condor Project
Computer Sciences Department
University of Wisconsin-Madison

You want Condor X.Y.Z

- But you don't want
  - Long down time
  - Killed jobs
  - Loss of configuration settings
  - Stale configuration settings
  - Surprises

Overview

- Package management
- Config file management
- Condor testing strategies
- Draining off jobs

Linux Packages

- Old Condor rpm
  - /opt/condor-x.y.z
  - Going away in 7.5.X
- New improved packages:
  - www.cs.wisc.edu/condor/yum
  - www.cs.wisc.edu/condor/debian

More packages

- Your linux distro may package Condor
  - e.g. Fedora, Debian
  - Some optional features may be disabled:
    - Globus, standard universe, …
- tarball or zip file
  - Useful for rolling your own package
  - Or installing on a shared file system
- source code

1st
A Concrete Example

yum

- See: www.cs.wisc.edu/condor/yum

yum update condor

- Are we done?

yum

- Doesn't touch modified config files
  - Don't edit /etc/condor/condor_config
    - defaults
  - Edit /etc/condor/condor_config.local
    - customization
  - Check release notes for recommended changes to your customized config settings

yum

- Does a fast shutdown of Condor
  - startd kills jobs immediately
  - drain jobs in advance if desired
- In future (7.4.3) will instead leave condor running
  - Condor will do graceful restart
  - Configuration can control whether jobs are killed

yum

- What else doesn't 'yum update condor' do?
  - pool-wide configuration management
  - testing
  - job drain-off (if desired)
  - control which machines update first

Configuration Management

condor_config

```
##  How long are you willing to let
##  daemons try their graceful
##  shutdown methods before they do a
##  hard shutdown? (30 minutes)
#SHUTDOWN_GRACEFUL_TIMEOUT      = 1800
```

- Most entries commented out with default value
- But some required settings are made
- Avoid editing this file

Dealing with a new config

- Diff base config with your config
- Understand new items
- Documented in manual version-history
- Existing ones rarely change
  - Usually capacity, not meaning changes
- Almost always, overwriting base file works

condor_config.local

- This file can point to additional customized config files via LOCAL_CONFIG_FILE
- Organize settings. Example:
  - condor_config.global
    - ALLOW_WRITE = *.cs.wisc.edu
  - condor_config.cm
    - DAEMON_LIST = MASTER, COLLECTOR, NEGOTIATOR
  - condor_config.submit
    - DAEMON_LIST = MASTER, SCHEDD
  - condor_config.execute
    - DAEMON_LIST = MASTER, STARTD

Configuration management

- Many possibilities
  - ROCKS, cfengine, Cycle Server, ZenWorks, Shared FS
- Example:
  - copy custom config files to all nodes
  - only condor_config.local differs
    - LOCAL_CONFIG_FILE = condor_config.global, condor_config.cm
    - LOCAL_CONFIG_FILE = condor_config.global, condor_config.submit

Incremental testing!

- Three basic components of Condor:
  - Central Manager
  - Submit points
  - Execute machines
- Can test each independently
  - Before or during upgrade

Compatibility Guarantees

- Can part of pool run old Condor and part run new Condor?
- No guarantees…
  - Check release notes
- But we try very hard!
  - Both forward and backward
- Flocking requires this

Testing Central Manager

- If it breaks, existing jobs keep running
- What I do: update the real CM
- More cautious: update HAD CM
  - Temporarily stop main CM
- Observe updated CM match jobs to machines (NegotiatorLog)

Testing submit machine

- Adding a new test schedd is easy
  - submit jobs, watch them succeed
  - if possible, run a real workflow
- Upgrading a real schedd
  - Std universe jobs checkpoint
  - Others can continue running
    - Default JobLeaseDuration is 20 minutes

Testing execute machine

- Can usually afford to upgrade one or more real execute nodes
  - verify that jobs run successfully
  - submit jobs from new schedd

Independent Testbed

- Extra cautious approach
- Create independent pool
  - Some options:
    - VMs
    - relocatable rpms on same host (or tarball)
    - Drain off part of main pool and repurpose machines
- Test real workflows, run benchmarks

Draining Jobs

- To drain or not to drain
  - Want minimal work loss
  - But maximum throughput
    - Some cores idle while others finish jobs
  - Checkpointable jobs less of a problem
    - But beware of overwhelming checkpoint storage server!

Draining Jobs

- See the How-to: HowToShutDownCondor

condor_off -all -startd –peaceful

- Once condor_status is empty, upgrade

Draining Jobs

- Don't want to wait for peaceful shutdown?
- Configure:
  MaxJobRetirementTime = 24*3600
  SHUTDOWN_GRACEFUL_TIMEOUT = 24*3600
- Upgrade
  - condor_master will do graceful restart
  - Note: broken in current rpm, to be fixed in 7.4.3

Standard Universe

- More sensitive to backward compatibility
- Job's LastCheckpointPlatform must match machine's CheckpointPlatform
- Checkpoint platform may change
  - On Condor upgrade
  - On OS upgrade

Draining Std Universe Jobs

- Some users have multi-month std universe jobs!
- Keep a few old startds around
  - To finish old standard uni jobs
  - Set START to "JobUniverse == 1"
  - Or maybe rank…

Big bang approach

- What we do at UW CS
- Just change a symlink to the binaries (in AFS)
  - Masters will notice updated binaries and restart

Incremental update

- First, update CM
  - No jobs lost
- Next, update schedd(s)
  - If restart happens in 20 minutes, jobs keep running
- Next, update startds

When to upgrade?

- Zeroth law of software engineering
- Development series actually pretty stable
- We'll let you know about security issues
- Probably don't need every minor version
- Don't be more than one major stable version behind

In summary…

- Pick a package/config manager
- Organize config files
- Test each component
- Drain jobs if desired