The next 60 minutes...

- Condor Daemons & Job Startup
- Configuration Files
- Policy Expressions
  - Startd (Machine)
  - Negotiator
- Priorities
- Security, briefly
- Useful Tools
- Log Files
- Debugging Jobs
Configuration Files

“amp wiring” by “fbz_” © 2005
Licensed under the Creative Commons Attribution 2.0 license
http://www.flickr.com/photos/fbz/114422787/
Configuration File

- Found either in file pointed to with the CONDOR_CONFIG environment variable, /etc/condor/condor_config, or ~condor/condor_config
- All settings can be in this one file
- Might want to share between all machines (NFS, automated copies, Wallaby, etc)
Other Configuration Files

- LOCAL_CONFIG_FILE setting
  - Comma separated, processed in order
  ```
  LOCAL_CONFIG_FILE = \\
  /var/condor/config.local,\\
  /var/condor/policy.local,\\
  /shared/condor/config.$(HOSTNAME),\\
  /shared/condor/config.$(OPSYS)
  ```
Configuration File Syntax

# I’m a comment!
CREATE_CORE_FILES=TRUE
MAX_JOBS_RUNNING = 50
# Condor ignores case:
log=/var/log/condor
# Long entries:
collector_host=condor.cs.wisc.edu,\secondary.cs.wisc.edu
Configuration File Macros

› You reference other macros (settings) with:
  • A = $(B)
  • SCHEDD = $(SBIN)/condor_schedd

› Can create additional macros for organizational purposes
Configuration File Macros

› Can append to macros:
  A=abc
  A=\$(A),def

› Don’t let macros recursively define each other!
  A=\$(B)
  B=\$(A)
Configuration File Macros

Later macros in a file overwrite earlier ones

- B will evaluate to 2:
  A=1
  B=${ (A)}
  A=2
Macros and Expressions Gotcha

› These are simple replacement macros
› Put parentheses around expressions

TEN=5+5
HUNDRED=$(TEN)*$(TEN)
  • HUNDRED becomes 5+5*5+5 or 35!
TEN=(5+5)
HUNDRED=($(TEN)*$(TEN))
  • ((5+5)*(5+5)) = 100
"Don't even think about it" by Kat “tyger_lyllie” © 2005
Licensed under the Creative Commons Attribution 2.0 license
http://www.flickr.com/photos/tyger_lyllie/59207292/
http://www.webcitation.org/5XIh5mYGS
Policy

Who gets to run jobs, when?
Policy Expressions

- Specified in `condor_config`
  - Ends up `startd/machine ClassAd`
- Policy evaluates both a machine `ClassAd` and a job `ClassAd` together
  - Policy can reference items in either `ClassAd` (See manual for list)
- Can reference `condor_config` macros: `${MACRONAME}`
Machine (Startd) Policy Expressions

- START
- RANK
- SUSPEND
- CONTINUE
- PREEMPT
- KILL
START

- START is the primary policy
- When FALSE the machine enters the Owner state and will not run jobs
- Acts as the Requirements expression for the machine, the job must satisfy START
  - Can reference job ClassAd values including Owner and ImageSize
RANK

› Indicates which jobs a machine prefers
  • Jobs can also specify a rank

› Floating point number
  • Larger numbers are higher ranked
  • Typically evaluate attributes in the Job ClassAd
  • Typically use + instead of &&
RANK

- Often used to give priority to owner of a particular group of machines
- Claimed machines still advertise looking for higher ranked job to preempt the current job
SUSPEND and CONTINUE

› When SUSPEND becomes true, the job is suspended
› When CONTINUE becomes true a suspended job is released
PREEMPT and KILL

When PREEMPT becomes true, the job will be politely shut down
  • Vanilla universe jobs get SIGTERM
    • Or user requested signal
  • Standard universe jobs checkpoint

When KILL becomes true, the job is SIGKILLed
  • Checkpointing is aborted if started
Minimal Settings

- Always runs jobs
  - START = True
  - RANK =
  - SUSPEND = False
  - CONTINUE = True
  - PREEMPT = False
  - KILL = False
Policy Configuration

› I am adding nodes to the Cluster... but the Chemistry Department has priority on these nodes
New Settings for the Chemistry nodes

- Prefer Chemistry jobs
  
  \[
  \begin{align*}
  \text{START} & = \text{True} \\
  \text{RANK} & = \text{Department} == \"Chemistry\" \\
  \text{SUSPEND} & = \text{False} \\
  \text{CONTINUE} & = \text{True} \\
  \text{PREEMPT} & = \text{False} \\
  \text{KILL} & = \text{False}
  \end{align*}
  \]
Submit file with Custom Attribute

Prefix an entry with "+" to add to job ClassAd

Executable = charm-run
Universe = standard
+Department = "Chemistry"
queue
What if “Department” not specified?

START = True
RANK = Department != UNDEFINED \ & Department == "Chemistry"
SUSPEND = False
CONTINUE = True
PREEMPT = False
KILL = False
More Complex RANK

› Give the machine’s owners (adesmet and roy) highest priority, followed by the Chemistry department, followed by the Physics department, followed by everyone else.

• Can use automatic Owner attribute in job attribute to identify adesmet and roy
More Complex RANK

\begin{verbatim}
IsOwner = (Owner == "adesmet" \ 
    || Owner == "roy")
IsChem = (Department != UNDEFINED \ 
    && Department == "Chemistry")
IsPhys = (Department != UNDEFINED \ 
    && Department == "Physics")
RANK = $(IsOwner)*20 + $(IsChem)*10 \ 
    + $(IsPhys)
\end{verbatim}
I have an unhealthy fixation with PBS so... kill jobs after 12 hours, except Physics jobs get 24 hours.
Useful Attributes

› **CurrentTime**
  • *Current time, in Unix epoch time (seconds since midnight Jan 1, 1970)*

› **EnteredCurrentActivity**
  • *When did Condor enter the current activity, in Unix epoch time*
Configuration

ActivityTimer = \n    (CurrentTime - EnteredCurrentActivity)
HOUR = (60*60)
HALFDAY = ($(HOUR)*12)
FULLDAY = ($(HOUR)*24)
PREEMPT = \n    ($(IsPhys) && ($(ActivityTimer) > $FULLDAY)) \n     || \n     (!$(IsPhys) && ($(ActivityTimer) > $HALFDAY))
KILL = $(PREEMPT)
Cluster is okay, but...

Condor can only use the desktops when they would otherwise be idle
Defining Idle

One possible definition:

- No keyboard or mouse activity for 5 minutes
- Load average below 0.3
Desksops should

- START jobs when the machine becomes idle
- SUSPEND jobs as soon as activity is detected
- PREEMPT jobs if the activity continues for 5 minutes or more
- KILL jobs if they take more than 5 minutes to preempt
Useful Attributes

› LoadAvg
  • Current load average

› CondorLoadAvg
  • Current load average generated by Condor

› KeyboardIdle
  • Seconds since last keyboard or mouse activity
Macros in Configuration Files

NonCondorLoadAvg = (LoadAvg - CondorLoadAvg)
BgndLoad = 0.3
CPU_Busy = ($(NonCondorLoadAvg) >= $(BgndLoad))
CPU_Idle = ($(NonCondorLoadAvg) < $(BgndLoad))
KeyboardBusy = (KeyboardIdle < 10)
KeyboardIsIdle = (KeyboardIdle > 300)
MachineBusy = ($(CPU_Busy) || $(KeyboardBusy))
Desktop Machine Policy

START = $(CPU_Idle) && $(KeyboardIsIdle)
SUSPEND = $(MachineBusy)
CONTINUE = $(CPU_Idle) && KeyboardIdle > 120
PREEMPT = (Activity == "Suspended") && \ $(ActivityTimer) > 300
KILL = $(ActivityTimer) > 300
Mission Accomplished

Smiles and kittens for everyone!

"Autumn and Blue Eyes" by Paul Lewis ("PJLewis") © 2005 Licensed under the Creative Commons Attribution 2.0 license
See the manual for the gory details. (Section 3.5: Policy Configuration for the condor_startd)
Custom Machine Attributes

Can add attributes to a machine’s ClassAd, typically done in the local configuration file

```
INSTRUCTIONAL=TRUE
NETWORK_SPEED=1000
STARTD_EXPRS=INSTRUCTIONAL, NETWORK_SPEED
```
Custom Machine Attributes

› Jobs can now specify Rank and Requirements using new attributes:

   Requirements = \n
   (INSTRUCTIONAL=?=UNDEFINED \n    || INSTRUCTIONAL==FALSE)

   Rank = NETWORK_SPEED

› Dynamic attributes are available; see STARTD_CRON_* settings in the manual
Further Machine Policy Information

› For further information, see section 3.5 “Policy Configuration for the condor_startd” in the Condor manual

› condor-users mailing list
  http://www.cs.wisc.edu/condor/mail-lists/

› condor-admin@cs.wisc.edu
Job Priority

- Set with `condor_prio`
- Users can set priority of their own jobs
- Integers, larger numbers are higher priority
- Only impacts order between jobs for a single user on a single schedd
- A tool for users to sort their own jobs
User Priority

- Determines allocation of machines to waiting users
- View with `condor_userprio`
- Inversely related to machines allocated (lower is better priority)
  - A user with priority of 10 will be able to claim twice as many machines as a user with priority 20
User Priority

Effective User Priority is determined by multiplying two components:

- Real Priority
- Priority Factor
Real Priority

- Based on actual usage
- Defaults to 0.5
- Approaches actual number of machines used over time
  - Configuration setting
    PRIORITY_HALFLIFE
Priority Factor

- Assigned by administrator
  - Set with condor_userprio
- Defaults to 1 (DEFAULT_PRIO_FACTOR)
Negotiator Policy Expressions

› **PREEMPTION_REQUIREMENTS** and **PREEMPTION_RANK**

› Evaluated when `condor_negotiator` considers replacing a lower priority job with a higher priority job

› Completely unrelated to the **PREEMPT** expression
PREEMPTION_REQUIREMENTS

› If false will not preempt machine
  • Typically used to avoid pool thrashing
  • Typically use:
    • RemoteUserPrio - Priority of user of currently running job (higher is worse)
    • SubmitterPrio - Priority of user of higher priority idle job (higher is worse)

› PREEMPTION_REQUIREMENTS=FALSE
PREEMPTION_REQUIREMENTS

Only replace jobs running for at least one hour and 20% lower priority

StateTimer = \ 
CurrentTime - EnteredCurrentState
HOUR = (60*60)
PREEMPTION_REQUIREMENTS = \ 
$(StateTimer) > (1 * $(HOUR)) \ 
&& RemoteUserPrio > SubmittorPrio * 1.2
PREEMPTION_RANK

- Picks which already claimed machine to reclaim
- Strongly prefer preempting jobs with a large (bad) priority and a small image size

PREEMPTION_RANK = \( \text{RemoteUserPrio} \times 1000000 \) - ImageSize
Security, briefly
Condor Security

› Strong authentication of users and daemons
› Encryption over the network
› Integrity checking over the network
Minimal Security Settings

› You must set ALLOW_WRITE, or nothing works
› Simplest setting:
  ALLOW_WRITE=*  
   • Extremely insecure!
› A bit better:
  ALLOW_WRITE= \
   * . cs . wisc . edu
More on Security

› Zach’s talk later today!
› Chapter 3.6, “Security,” in the Condor Manual
› condor-admin@cs.wisc.edu
condor_config_val

› Find current configuration values
% condor_config_val MASTER_LOG
/var/condor/logs/MasterLog
% cd `condor_config_val LOG`
condor_config_val -v

Can identify source

% condor_config_val -v CONDOR_HOST
CONDOR_HOST: condor.cs.wisc.edu

Defined in ‘/etc/
condor_config.hosts’, line 6
What configuration files are being used?

% condor_config_val -config
Config source:
  /var/home/condor/condor_config
Local config sources:
  /unsup/condor/etc/condor_config.hosts
  /unsup/condor/etc/condor_config.global
  /unsup/condor/etc/condor_config.policy
  /unsup/condor-test/etc/hosts/puffin.local
condor_fetchlog

› Retrieve logs remotely

condor_fetchlog  
beak.cs.wisc.edu Master
Querying daemons

condor_status

Queries the collector for information about daemons in your pool

- Defaults to finding condor_startds

- `condor_status -schedd` summarizes all job queues

- `condor_status -master` returns list of all condor_masters
condor_status

- `long` displays the full ClassAd
- Optionally specify a machine name to limit results to a single host

```
condor_status -l
node4.cs.wisc.edu
```
condor_status -constraint

▷ Only return ClassAds that match an expression you specify
▷ Show me idle machines with 1GB or more memory

• `condor_status -constraint 'Memory >= 1024 && Activity == "Idle"'`
condor_status -format

› Controls format of output
› Useful for writing scripts
› Uses C printf style formats
  • One field per argument
condor_status -format

• Census of systems in your pool:

% condor_status -format '%s ' Arch -format '%s\n' OpSys |
sort | uniq -c

  797 INTEL LINUX
  118 INTEL WINNT50
  108 SUN4u SOLARIS28
  6 SUN4x SOLARIS28
Examining Queues `condor_q`

- View the job queue
- The “-long” option is useful to see the entire ClassAd for a given job
- Supports `-constraint` and `-format`
- Can view job queues on remote machines with the “-name” option
condor_q -format

> Census of jobs per user
% condor_q -format '%8s ' Owner
  -format '%s\n' Cmd | sort |
  uniq -c

  64 adesmet /scratch/submit/a.out
  2 adesmet /home/bin/run_events
  4 smith /nfs/sim1/em2d3d
  4 smith /nfs/sim2/em2d3d
condor_q -analyze

› condor_q will try to figure out why the job isn’t running
› Good at determining that no machine matches the job Requirements expressions
condor_q -analyze

Typical intro:

% condor_q -analyze 471216
471216.000: Run analysis summary. Of 820 machines,
  458 are rejected by your job's requirements
  25 reject your job because of their own requirements
  0 match, but are serving users with a better priority in the pool
  4 match, but reject the job for unknown reasons
  6 match, but will not currently preempt their existing job
  327 are available to run your job
    Last successful match: Sun Apr 27 14:32:07 2008
condor_q -analyze

Continued, and heavily truncated:

The Requirements expression for your job is:

```
( ( target.Arch == "SUN4u" ) && ( target.OpSys == "WINNT50" ) && [snip]
```

<table>
<thead>
<tr>
<th>Condition</th>
<th>Machines</th>
<th>Suggestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (target.Disk &gt; 100000000)</td>
<td>0</td>
<td>MODIFY TO 14223201</td>
</tr>
<tr>
<td>2 (target.Memory &gt; 10000)</td>
<td>0</td>
<td>MODIFY TO 2047</td>
</tr>
<tr>
<td>3 (target.Arch == &quot;SUN4u&quot;)</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>4 (target.OpSys == &quot;WINNT50&quot;)</td>
<td>110</td>
<td>MOD TO &quot;SOLARIS28&quot;</td>
</tr>
</tbody>
</table>

Conflicts: conditions: 3, 4
Condor’s Log Files

- Condor maintains one log file per daemon
- Can increase verbosity of logs on a per daemon basis
  - `SHADOW_DEBUG`, `SCHEDD_DEBUG`, and others
  - Space separated list
Useful Debug Levels

- **D_FULLDEBUG** dramatically increases information logged
  - Does not include other debug levels!
- **D_COMMAND** adds information about commands received

```c
SHADOW_DEBUG = D_FULLDEBUG D_COMMAND
```
Log Rotation

Log files are automatically rolled over when a size limit is reached

• Only one old version is kept
• Defaults to 1,000,000 bytes
• Rolls over quickly with D_FULLDEBUG
• MAX_*_LOG, one setting per daemon
  • MAX_SHADOW_LOG, MAX_SCHEDD_LOG, and others
Condor’s Log Files

Many log files entries primarily useful to Condor developers

- Especially if D_FULLDEBUG is on
- Minor errors are often logged but corrected
- Take them with a grain of salt
- condor-admin@cs.wisc.edu
Debugging Jobs: 
condor_q

- Examine the job with condor_q
  - especially -long and -analyze
  - Compare with condor_status -long for a machine you expected to match
Debugging Jobs: User Log

› Examine the job’s user log
  • Can find with:
    condor_q -format '%s\n' UserLog 17.0
  • Set with “log” in the submit file
  • You can set EVENT_LOG to get a unified log for all jobs under a schedd

› Contains the life history of the job

› Often contains details on problems
Debugging Jobs: ShadowLog

› Examine ShadowLog on the submit machine
  • Note any machines the job tried to execute on
  • There is often an “ERROR” entry that can give a good indication of what failed
Debugging Jobs: Matching Problems

- No ShadowLog entries? Possible problem matching the job.
  - Examine ScheddLog on the submit machine
  - Examine NegotiatorLog on the central manager
Debugging Jobs: Remote Problems

- ShadowLog entries suggest an error but aren’t specific?
  - Examine StartLog and StarterLog on the execute machine
Debugging Jobs: Reading Log Files

- Condor logs will note the job ID each entry is for
  - Useful if multiple jobs are being processed simultaneously
  - grepping for the job ID will make it easy to find relevant entries
Debugging Jobs: What Next?

› If necessary add “D_FULLDEBUG D_COMMAND” to DEBUG_DAEMONNAME setting for additional log information

› Increase MAX_DAEMONNAME_LOG if logs are rolling over too quickly

› If all else fails, email us
  • condor-admin@cs.wisc.edu
More Information

› Condor staff here at Condor Week
› Condor Manual
› condor-users mailing list
  http://www.cs.wisc.edu/condor/mail-lists/
› condor-admin
  condor-admin@cs.wisc.edu

“Condor Manual” by Alan De Smet
(Actual first page of the 7.0.1 manual on about 700 pages of other output. The actual 7.0.1 manual is about 860 pages.)
Thank You!

Any questions?

“My mouse” by “MysterFaery” © 2006 Licensed under the Creative Commons Attribution 2.0 license