Things you may not know about HTCondor

John (TJ) Knoeller

Condor Week 2017

-limit not just for condor_history

`condor_q -limit <num>`

- Show no more than <num> jobs.
  - Ignored if Schedd is before 8.6

`condor_status -limit <num>`

- Show no more than <num> ads
  - Ignored if Collector is before version 8.7.1

You can use -af to test classad expressions

```
>condor_status -limit 1 -af 'subsys("abcdef",1,-2)'
bcd

>condor_history -lim 1 -af 'split("a,b,c")'
{"a","b","c"}

>condor_q -lim 1 -af:V 'join("/",split("a,b,c"))'
a/b/c
```

You can use the classad ?: operator as a binary operator

- In 8.6 the middle argument of the ?: operator is now optional in -format, -af, and custom print format files

```
# if you want this
Attr =!= undefined ? Attr : Default

# you can write this!
(Attr ?: Default)
```

- Be sure to use () around it!
- Not safe for use in Job or Machine ads (yet)
  - An 8.4 or earlier Collector / Schedd will not read it

Print format files configure the classad pretty-printer

Save this as job_starts.fmt

```
SELECT
  Machine WIDTH -12
  splitslotname(Name)[0] AS Slot WIDTH -5
  Cpus
  Memory
  JobId?:"no" AS JobId WIDTH -5
  RecentJobStarts/20.0 AS 'Jobs/Min' PRINTF %.2f
WHERE RecentJobStarts >= 1
```

```
>condor_status -pr job_starts.fmt
Machine     Slot  Cpus Memory JobId Jobs/Min
bob.cs.edu slot1     1    256  27.2     0.15
```

You can set the default output of condor_status and condor_q

```
STATUS_DEFAULT_<type>_PRINT_FORMAT_FILE \
  = /path/to/print-format-file
<type> is STARTD, SCHEDD, MASTER, etc

Q_DEFAULT_PRINT_FORMAT_FILE = /path/to/file
  or
Q_DEFAULT_<opt>_PRINT_FORMAT_FILE = /path/to
<opt> is RUN, HOLD, GRID, etc
```

Submit files now have keywords for controlling job retries

`max_retries = <num>`

- Retry the job until it exits with a success exit code or <num> retries have happened

`success_exit_code = <value>`

- Set the success value for the exit code. Default is 0

`retry_until = <expression>`

- Retry the job until <expression> is true or <num> retries have happened.

$() expansions in config and submit files allow defaults

- Use in config file to safely amend START expression

```
START = $(START:true) && TARGET.WantMore
```

- Use in sleep.submit file to allow command line values

```
Executable = /bin/sleep
Args = $(SLEEP_FOR:300)
```

```
>condor_submit SLEEP_FOR=5 sleep.submit
```

sleeps for 5 seconds

```
>condor_submit sleep.submit
```

sleeps for 300 seconds

In 8.6 $ENV also allows default

```
+SubmittedBy="$ENV(LOGNAME:nobody)"
```

- $ENV macro now accepts a default value

```
$Fpduwnxbqa(param)
```

- $F macro has additional formatting options
  - u and w format paths as *nix and Windows.
  - b strips trailing / from paths and . from ext
  - qa quotes for use in Arguments

You can use if and include in config files

```
if $(IsMaster)
  include command into $(CONFIG_DIR)/cache : <script>
else
  include ifexist : $(CONFIG_DIR)/cache
endif
```

- Builtin IsXXX params for each deamon type
- Include can read a file or run a script and read its output
- Optional keywords in green are new for 8.6

You can use if and include in submit files also

```
include ifexist : boilerplate.submit
if defined OPTION
  Args = $(Args) -opt $(OPTION)
endif
```

- The option argument can be passed on the command line
- if defined is true if the option is defined and not empty

Metaknobs are now called "Configuration Templates"

- Simplify configuration
- Future proof
- Four categories
  - Role
  - Security
  - Feature
  - Policy

condor_config_val will list the config templates

```
> condor_config_val use FEATURE
use FEATURE accepts
  ContinuousCronHook
  GPUs
  OneShotCronHook
  PartitionableSlot
  PeriodicCronHook
  REMOTE_CONFIG
  REMOTE_RUNTIME_CONFIG
  ScheddCronContinuous
  ScheddCronOneShot
  ScheddCronPeriodic
  StartdCronContinuous
  StartdCronOneShot
  StartdCronPeriodic
  VMWARE
```

```
> condor_config_val use POLICY
use POLICY accepts
  ALWAYS_RUN_JOBS
  DESKTOP
  HOLD_IF_CPUS_EXCEEDED
  HOLD_IF_MEMORY_EXCEEDED
  LIMIT_JOB_RUNTIMES
  PREEMPT_IF
  PREEMPT_IF_CPUS_EXCEEDED
  PREEMPT_IF_MEMORY_EXCEEDED
  UWCS_DESKTOP
  WANT_HOLD_IF
```

Metaknobs with arguments

`use FEATURE : PartitionableSlot(1, Cpus=2, Memory=50%)`

- Register a type 1 partitionable slot and give it 2 Cpus and 50% of the memory

`use FEATURE : StartdCronOneShot(<tag>,<exe>[,<args>])`

- Register a One-shot STARTD_CRON hook, using <tag> as the hook name, and <exe> as the program.

`use FEATURE : StartdCronPeriodic(<t>,<rate>,<e>[,<a>])`

- Register a Periodic STARTD_CRON hook, using <t> as the hook name, rate as the repeat rate and <e> as the program.

condor_config_val tricks

`condor_config_val -write:upgrade -`

- Show all non-default values in the config files

`condor_config_val -subsys SCHEDD`

- Parse the config files like the schedd would

`condor_config_val -dump _LOG$`

- Show all values with names that end in _LOG

Config trick #1

```
include : /bin/echo RANDVAL=$RANDOM_INTEGER(1,9) |
```

- RANDVAL changes only on startup and on reconfig of the daemon

- Why does this work?

Config trick #2

```
include command into $(LOCAL_CONFIG_DIR)/randv : \
/bin/echo RANDVAL=$RANDOM_INTEGER(10000,99999)
```

- The first parse of config sets a random value into a file, all others parse the file.

- ***THIS HAS SECURITY IMPLICATIONS!***
  - If the file randv is writable by non-root, an attacker can use it to get root.

CENTER FOR
HIGH THROUGHPUT
COMPUTING

HTCondor

Any Questions?