Transforming Jobs in the Schedd

John (TJ) Knoeller
Center for High Throughput Computing

Motivation

You want to have a policy about what jobs are allowed, to require certain attributes, or to make it easy for users

- Submit requirements
- Submit attributes
- Job transforms
- Preventing subsequent changes

Enforce a job policy or schema

Policy: All jobs must have **`Experiment`** attribute
Reject jobs at submit time that don't have this attribute

```
SUBMIT_REQUIREMENT_NAMES = $(SUBMIT_REQUIREMENT_NAMES) CheckExp
SUBMIT_REQUIREMENT_CheckExp = JobUniverse == 7 || Experiment isnt undefined
SUBMIT_REQUIREMENT_CheckExp_REASON = "submissions must have +Experiment"

# JobUniverse 7 is Scheduler universe, i.e. DAGMAN.
# JobUniverse 12 is Local universe, maybe ignore these jobs also?
```

Defaulting job attributes

Configure SUBMIT_ATTRS to add attributes to jobs.

```
SUBMIT_ATTRS = $(SUBMIT_ATTRS) Experiment
Experiment = "CHTC"
```

Job ClassAd starts with **`Experiment="CHTC"`**
Before the submit file is even processed

SUBMIT_ATTRS is weak

Good for setting defaults

Work happens outside of the SCHEDD

-however-

User can override or un-configure

Applies to ALL job types (Grid, VM, DAG)

May not happen with remote submit

Job Transforms in the Schedd

Configure JOB_TRANSFORM_*

```
JOB_TRANSFORM_NAMES = $(JOB_TRANSFORM_NAMES) SetExperiment
JOB_TRANSFORM_SetExperiment = [ set_Experiment = "CHTC"; ]
```

`Experiment="CHTC"` written into each job ClassAd as it is submitted, *overriding what user specified!*

Transforms can be conditional

```
JOB_TRANSFORM_NAMES = $(JOB_TRANSFORM_NAMES) SetExperiment
JOB_TRANSFORM_SetExperiment @=end
[
  Requirements = JobUniverse != 7 && Experiment is undefined;
  set_Experiment = "CHTC";
]
@end
```

Adds **`Experiment="CHTC"`** to each job that doesn't already have that attribute

Job transforms are classads (<9.0)

Job transforms are ClassAds in "new" syntax

```
[  Requirements = <job-match-constraint>;
<verb>_<subject-attr> = object;
 … ]
```

Verbs are **`copy`**, **`delete`**, **`set`** & **`eval_set`**

- Evaluated in that order, **`copy`** before **`delete`** before **`set`**, etc
- Evaluation order within each verb group is *random*

Job transforms look like:

```
JOB_TRANSFORM_Example @=end
[
   Requirements = JobUniverse == 5; /* vanilla jobs only */
   copy_FromAttr = "ToAttr";
   delete_Attr = true;
   set_StringAttr = "A";
   set_IntAttr = 100;
   set_Requirements = Arch != "ARM"; /* replace job req */
   eval_set_EnvA = join("=", StringAttr, "B"); /* "A=B" */
   eval_set_Environment = join(" ", Environment, EnvA);
]
@end
```

(there's a problem with the last 2 lines - remember eval_set order is *random*)

The motivation for transforms

How do I assign jobs to accounting groups automatically, while preventing users from cheating?

Job transforms + Immutable attributes

But doing this in ClassAd language is *painful*

```
eval_set_AcctGroup =
  IfThenElse(Owner=="Bob","CHTC",
    IfThenElse(Owner=="Alice","Math",
      IfThenElse(Owner=="Al","Physics","Unknown")
  ))
```

There a (meta) knob for that

```
use FEATURE : AssignAccountingGroup(/path/to/map)
```

You can run

```
condor_config_val use feature:AssignAccountingGroup
```

to see what this configuration template expands to

(*/path/to/map. what's that?)*

ClassAd map files

Map file is text, with 3 fields per line

* *<key_or_regex> <result_list>*

```
* Bob       CHTC,Security
* Alice     CHTC,Math,Physics
* /.*Hat/i  Problem
* /.*/      CHTC
```

The first field is a namespace, * for ‘default’

Configuring a map

```
CLASSAD_USER_MAPFILE_MyMap = /path/to/mapfile
            <or>
CLASSAD_USER_MAPDATA_MyMap @=end
 * Bob        CHTC,Security
 * Alice      CHTC,Math,Physics
 * /.*Hat$/i  Collab
 * /.*/       General
@end

# control which damons (or tools) load the map
SCHEDD_CLASSAD_USER_MAP_NAMES = MyMap $(SCHEDD_CLASSAD_USER_MAP_NAMES)
# TOOL_CLASSAD_USER_MAP_NAMES = MyMap $(TOOL_CLASSAD_USER_MAP_NAMES)
```

Use `userMap("MyMap",…)` in Classad expressions in the SCHEDD

userMap has 3 variants

```
result = userMap(mapname, input)
```

map input to all results (returns the list)

```
result = userMap(mapname, input, preferred)
```

map input to preferred or undefined

```
result = userMap(mapname, input, pref, def)
```

map input to preferred or default result

Preventing change

IMMUTABLE_JOB_ATTRS - Cannot be changed once set

PROTECTED_JOB_ATTRS - Cannot be changed by user

SECURE_JOB_ATTRS - Can only be changed by Security

```
IMMUTABLE_JOB_ATTRS = $(IMMUTABLE_JOB_ATTRS) Experiment
```

Putting it all together

```
SCHEDD_CLASSAD_USER_MAP_NAMES = GroupFromOwner $(SCHEDD_CLASSAD_USER_MAP_NAMES)
CLASSAD_USER_MAPFILE_GroupFromOwner = /path/to/mapfile

# Assign Accounting groups automatically based on submitting username (Owner attribute)
JOB_TRANSFORM_NAMES = AssignGroup $(JOB_TRANSFORM_NAMES)
JOB_TRANSFORM_AssignGroup @=end
[
   copy_AcctGroup = "RequestedGroup";
   eval_set_AccountingGroup = join(".", usermap("GroupFromOwner", Owner, RequestedGroup), Owner);
   eval_set_AcctGroup = usermap("GroupFromOwner", Owner, RequestedGroup);
]
@end

IMMUTABLE_JOB_ATTRS = $(IMMUTABLE_JOB_ATTRS) AcctGroup AcctGroupUser AccountingGroup
SUBMIT_REQUIREMENT_NAMES = $(SUBMIT_REQUIREMENT_NAMES) CheckGroup
SUBMIT_REQUIREMENT_CheckGroup = AcctGroup isnt undefined && AccountingGroup isnt undefined
SUBMIT_REQUIREMENT_CheckGroup_REASON = strcat("Could not map '", Owner, "' to a group")
```

New Job transform syntax in 9.0

Native syntax is similar to config/submit syntax

- Statements evaluated in declaration order
- Temporary variables
- If/then/else
- Rename and Delete by attribute name pattern

Transforms are converted from ClassAd syntax on load

9.0 transform example

```
# Use job transform to add a pool constraint to vanilla jobs
# based on whether the job needs GPUs or not
#
JOB_TRANSFORM_GPUS @=end
  REQUIREMENTS JobUniverse == 5
  tmp.NeedsGpus = $(MY.RequestGPUs:0) > 0
  if $INT(tmp.NeedsGpus)
    SET Requirements $(MY.Requirements) && (Pool == "ICECUBE")
  else
    SET Requirements $(MY.Requirements) && (Pool == "CHTC")
  endif
@end
```

Any Questions?