

HTCondor-CE 5 and Job Router Transforms

John (TJ) Knoeller, Brian Lin

Center for High Throughput Computing

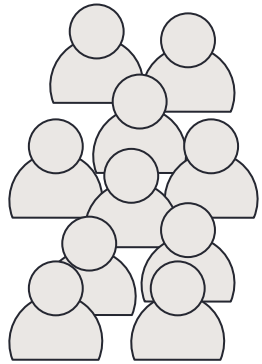
For a detailed introduction, see this
“HTCondor-CE Overview” talk

<https://indico.cern.ch/event/936993/contributions/4022131/>

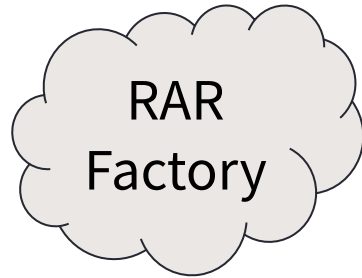
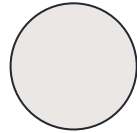
Or watch the YouTube video on our channel!

<https://www.youtube.com/watch?v=6IWaMbofG7M>

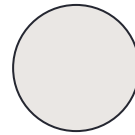
Resource Allocation Requests (RARs)



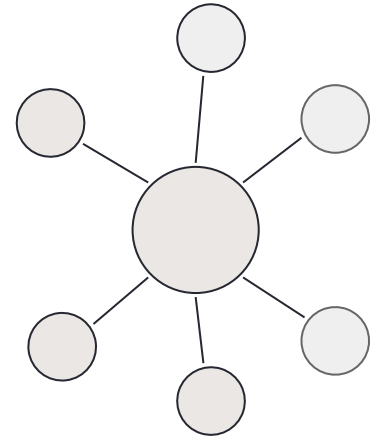
Access Point



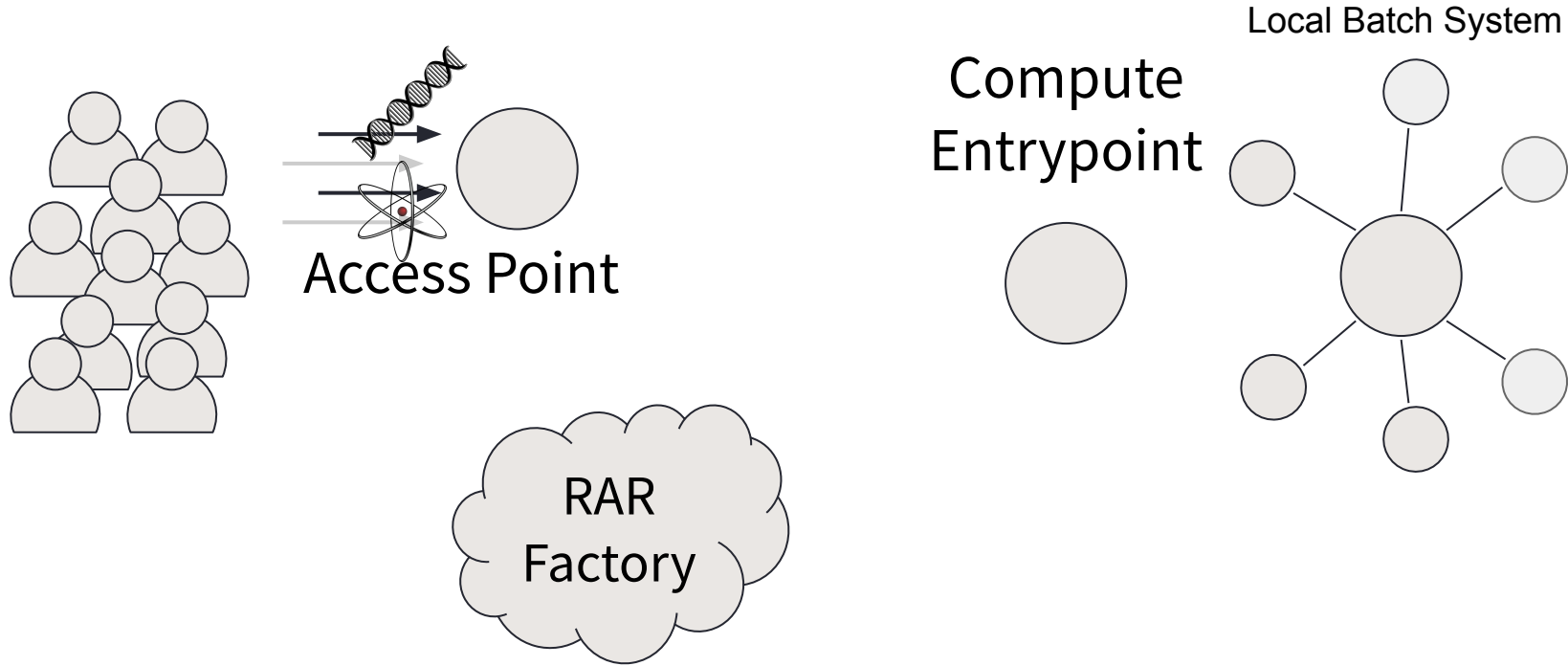
Compute
Entrypoint



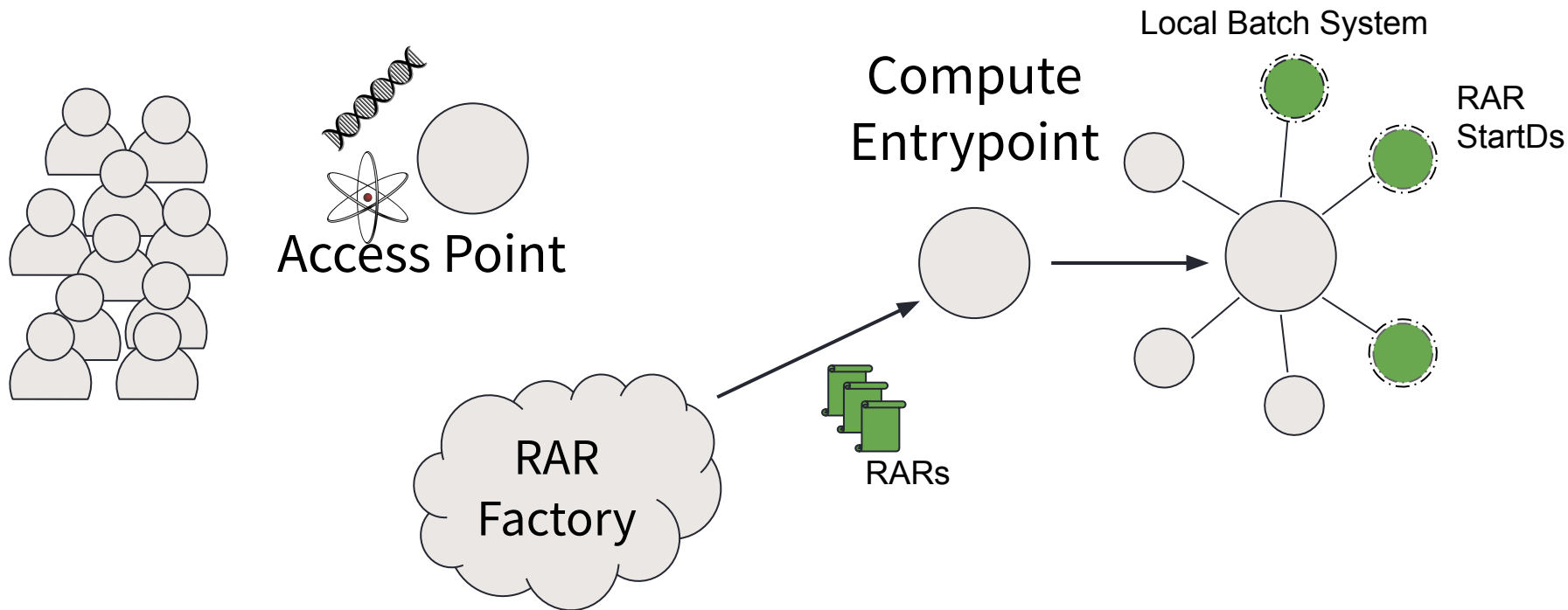
Local Batch System



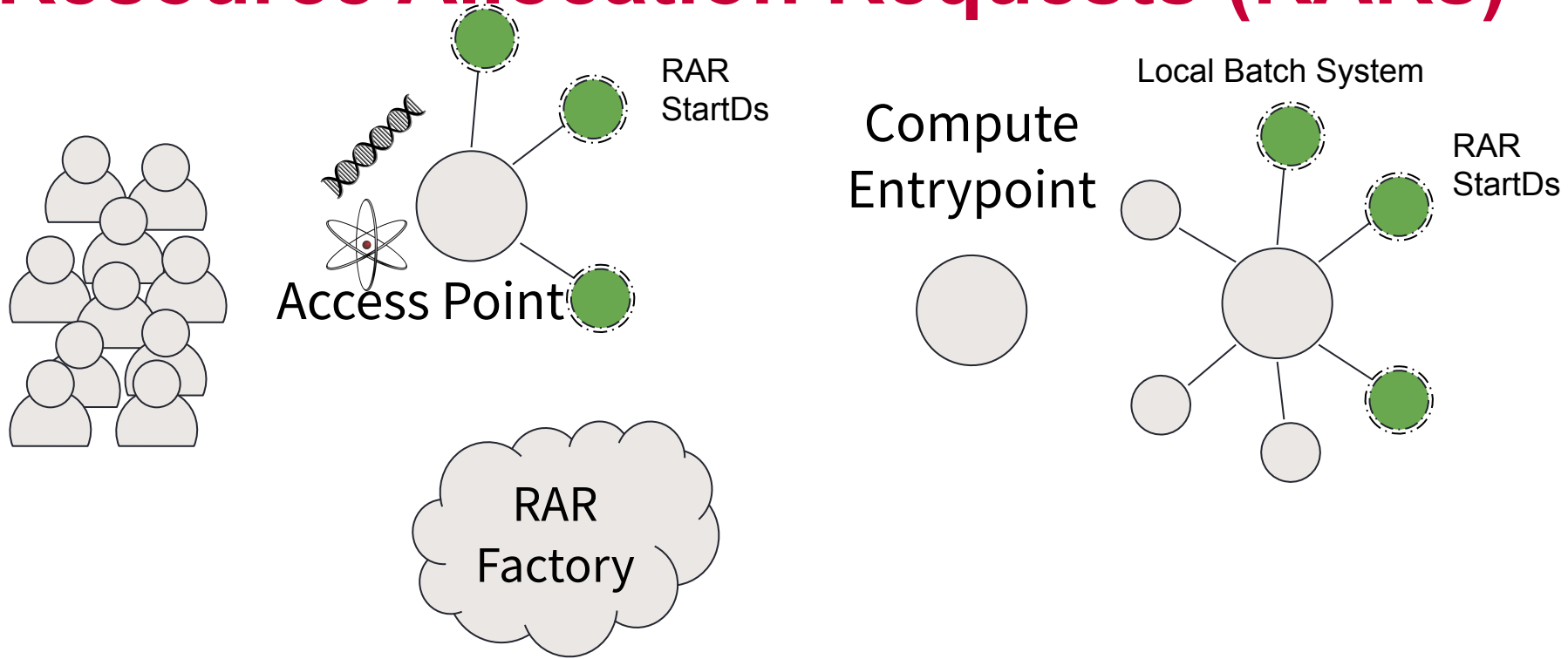
Resource Allocation Requests (RARs)



Resource Allocation Requests (RARs)



Resource Allocation Requests (RARs)



HTCondor-CE 5!

- The new major version available alongside the HTCondor 9.0 and 9.1 series
- Support added for an improved Job Router configuration syntax: Job Router Transforms
- `condor_mapfile` formatting and location changes
- Moved to Python 3 and added support for Enterprise Linux 8 based operating systems
- `$HOME` is no longer set by default in the RAR environment
- Upgrade instructions and detailed release notes available at the redesigned docs (<http://htcondor.org>): <https://htcondor.github.io/htcondor-ce/v5/releases/>

HTCondor-CE 5 Mapping Configuration

- CE administrators must now place their own RAR credential → local user mappings in a dedicated directory, `/etc/condor-ce/mapfiles.d/`
- Keep an eye out for `/etc/condor-ce/condor_mapfile.rpmsave`
- Downstream packagers can include default mappings in `/usr/share/condor-ce/mapfiles.d/`
- Credential matching strings must be converted to regular expressions
 - In the old syntax, a mapping based on the token issuer:
`SCITOKENS "https://scitokens.org/osg-connect" osgpilotuser`
 - In the new syntax:
`SCITOKENS /^https:\\/\\/scitokens\\.org\\/osg-connect,/ osgpilotuser`
 - Additionally, SciTokens/WLCG tokens can be mapped based on the token's subject (e.g., "testing"):
`SCITOKENS /^https:\\/\\/scitokens\\.org\\/osg-connect,testing$/ osgtestuser`

New Job Router config for HTCondor 9.0

Motivation

- Job Router use has changed
- Two different transform languages - Schedd submit transforms vs Job Router
 - Submit transforms are more powerful
 - Job router trying to do too much with a single transform.

Backward compatibility

- Pre HTCondor 9.0 configuration still works,
- Individual routes can use the new syntax
- If there is a route name conflict
 - old syntax route is ignored
 - logged in JobRouterLog on startup

Review : old config

- JOB_ROUTER_DEFAULTS
 - One huge ClassAd for all routes
- JOB_ROUTER_ENTRIES
 - List of ClassAds, one for each route
- Job Router splits the ENTRIES into individual route ads, then merges each with DEFAULTS ad(s) to get the effective route ads

New config splits things up

- One config knob for each route
- "defaults" is now pre and post route transforms
- Effective route is now a sequence
 - Zero or more pre-route transforms
 - One Route transform
 - Zero or more post-route transforms

New config knobs

JOB_ROUTER_ROUTE_NAMES

- list of route names to be matched in order

JOB_ROUTER_ROUTE_<name>

- defines a single route and match constraint

JOB_ROUTER_PRE_ROUTE_TRANSFORM_NAMES

JOB_ROUTER_POST_ROUTE_TRANSFORM_NAMES

- Applied to all jobs as they are routed

JOB_ROUTER_TRANSFORM_<name>

- Defines a single pre or post route transform

A route is a sequence

- Route REQUIREMENTS evaluated until a match is found
- A single route is applied:
 1. Create a temporary variable set
 2. Apply Pre-route transforms that match the job
 3. Apply the route
 4. Apply Post-route transforms that match the job
 5. Send the modified job classad to the destination schedd
 6. Clear the temporary variable set
- Each transform has access to changes to the job and variable set that are made by earlier transforms

Route/Transform commands

- Commands start with a keyword
- to modify the job

SET , DEFAULT , EVALSET

RENAME , COPY , DELETE

- or configure the route

REQUIREMENTS , UNIVERSE , NAME

- or set a temp variable

EVALMACRO

Route syntax similar to submit

name = value

- sets a temp variable - job is not changed
- temp variables used for \$() substitution
- cleared before the next job is routed
- a few are interpreted by the job router

if / else / endif

- used to make a block of statements conditional
- use EVALMACRO for complex conditionals

Some temp variables control routing

```
# Some special variable names configure the route
# These are read when the route is loaded
JOB_ROUTER_ROUTE_EX @=jre
  # if UNIVERSE is not used, then TargetUniverse is checked
  TargetUniverse = 5
  GridResource = "condor submit.area41.nv submit.area41.nv"
  MaxJobs = 100
  MaxIdleJobs = 50
  FailureRateThreshold = 0.03
  JobShouldBeSandboxed = <bool-expression>
  JobFailureTest = <bool-expression>
  EditJobInPlace = <bool-expression>
  UseSharedcx509UserProxy = <bool-expression>
  SharedX509UserProxy = <string-expression>

...
@jre
```

Route example

```
JOB_ROUTER_ROUTE_AREA41 @=jre
  UNIVERSE Grid
  REQUIREMENTS Project == "blue-book"

  # some temp variables in the route configure the router
  GridResource = "batch slurm"
  MaxIdleJobs = 20

  SET GPUNumber = 1
  SET ReferringSite = "$(MY_ROUTER_ID)"
@jre

# set config knobs that the route can use via $() substitution
MY_ROUTER_ID = wisc.edu
```

Pre Route transform example

```
# Example - Jobs that don't have a project defined, get a default
# project of "prj_<Owner>_<month>"

JOB_ROUTER_TRANSFORM_PROJECT @=jrt
    REQUIREMENTS Project is undefined
    EVALSET Project=join("_", "prj", Owner, formattime(qdate,"%b"))
@jrt

# This could also be written as

JOB_ROUTER_TRANSFORM_PROJECT @=jrt
    EVALMACRO tmp = join("_", "prj", Owner, formattime(qdate,"%b"))
    DEFAULT Project = "$ (tmp) "
@jrt
```

\$() substitution is ... *complicated*

- **\$(name)** - name is route temp variable OR condor config variable
- **\$(MY.name)** - name is a job attribute when there is a job, expands to nothing otherwise
- **\$()** substitution happens before evaluation
 - for a few variables when the route is loaded
 - for most variables when the route is used

ex: Modify job requirements

```
JOB_ROUTER_ROUTE_Foo @=jre
....
SET Requirements = ($(My.Requirements)) && site=="$(SITE)"
@jre
SITE = CHTC
```

Job ClassAd has

```
Requirements = Arch == "x86_64"
```

SET Statement

```
Requirements = ($(My.Requirements)) && site=="$(SITE)"
```

After \$() substitution

```
Requirements = (Arch == "x86_64") && site=="CHTC"
```

ex: Modify job environment

```
JOB_ROUTER_ROUTE_Foo @=jre
```

```
....
```

```
    EVALMACRO addenv = join(" ", "HOME=$(home)", "Y=$(INT(my.y))")
```

```
    SET Environment = "$(My.Environment) $(addenv)"
```

```
@jre
```

Job ClassAd has

```
Environment = "A=B"
```

```
y = 10 + 1
```

Transform Statements

```
addenv = join(" ", "HOME=$(HOME)", "Y=$(INT(my.y))")
```

```
Environment = "$(My.Environment) $(addenv)"
```

After \$() substitution

```
addenv = [eval] join(" ", "HOME=/usr/home/", "Y=11")
```

```
addenv = HOME=/usr/home Y=11
```

```
Environment = "A=B HOME=/user/home Y=11"
```

Testing your transforms

- `condor_job_router_info` (and `condor_ce_job_router_info`)
 - simulate routing of a job and see the result

```
condor_job_router_info -job <in> -route <out>
```

- `condor_transform_ads`
 - stand alone tool for testing a single transform
 - `-verbose` mode logs the transform steps

```
condor_transform_ads -rules <xfm> -in <in>
```

```
condor_transform_ads -help rules
```


Questions?

This material is based upon work supported by the National Science Foundation under Grant Nos. 1148698, 1321762, 1836650, and 2030508. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.