

REST API to HTCondor

Mátyás Selmecci

Today, you have two options to talk to HTCondor

- Option 1: Python bindings
 - Python only
 - Need compiled binaries for your platform

- Option 2: Command-line tools
 - Need to fork and exec other processes
 - Need compiled binaries for your platform

This leaves some developers out in the cold...

- Unsupported platforms
- Web developers
- Mobile developers

We need an interface anybody can use.

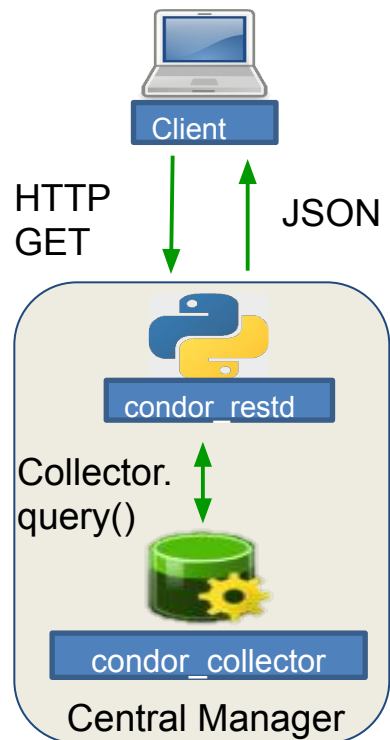
Solution: make a REST interface

- REST is a generic term for a software architecture whose properties include:
 - Client-server
 - Stateless (no persistent connection)
 - Uniform interface (all queries/commands look similar)
- In practice, this means HTTP; a REST interface will:
 - Listen to HTTP verbs GET, POST, etc.
 - Interpret URIs as queries or commands, not file paths
 - Respond with HTTP response codes (200, 404, etc.) and JSON (typically)

Benefits of REST

- Statelessness + client-server means you can treat queries as function calls
- Web servers take care of the networking bits
- Libraries widely available for most programming languages (even bash has curl and jq)

HTCondor-RESTD



- An 'adapter' between an HTTP client and HTCondor daemons
 - Translates HTTP queries into Python bindings calls
 - Translates responses into JSON
- Runs on the Central Manager (as a separate service)

Currently, the RESTD allows read-only queries

- GET /v1/jobs/... (condor_q)
- GET /v1/history/... (condor_history)
- GET /v1/status/... (condor_status)
- GET /v1/config/... (condor_config_val)

- Can give it constraints, projections (list of attributes to return)

- Note: done as the user running the RESTD

Example (JavaScript with jQuery)

```
<script
src="https://code.jquery.com/jquery-3.5.1.min.js">
</script>

<script>
$.getJSON( "http://localhost:9680/v1/status?query=startd" )
  .done(function( slots ) {

    slots.forEach(function(item) {

      var classad = item.classad;

      document.write( "name: " + item.name);
      document.write( " mem: " + classad.memory);
      document.write( " load: " + classad.loadavg);
      document.write( "<br>");

    });
  });
</script>
```

```
name: slot4@10af02a1d3e5 mem: 1808 load: 0
name: slot3@10af02a1d3e5 mem: 1808 load: 0
name: slot2@10af02a1d3e5 mem: 1808 load: 0
name: slot1@10af02a1d3e5 mem: 1808 load: 0.67
```

(see [full example](#))

Two ways to try it out

1. Install it with pip as a non-root user on an existing HTCondor host -- see [installation instructions](#).
2. Run the minicondor (htcondor/mini) container -- see [instructions](#).

Note: this is experimental software (APIs are subject to change); don't run it in production!

Future work

- Authentication and write operations
- (Good) Generated bindings for various languages
- This is experimental software -- we welcome your input!

Thanks and Links

- Greg Thain and the HTCondor Team
- Boris Sadkhin (Argonne National Laboratory)

- [htcondor-restd GitHub repository](#)