Gateway Production Monitoring

K. Yoshimoto
San Diego Supercomputer Center
University of California San Diego

May 19, 2020 / ECSS Symposium
1 Outline
- Introduction
- REST Interface
- Browser Interface
Outline

1 Outline
   - Introduction
   - REST Interface
   - Browser Interface
The Neuroscience Gateway (NSG) is a science gateway for computational neuroscience research. It is based on the Workbench Framework (WF) code originally used in the CIPRES Science Gateway at https://phylo.org. Another gateway using WF is COSMIC2 (https://cosmic-cryoem.org/), a cryoem gateway. The WF gateway app can provide both a non-API web browser interface and a RESTful API. NSG makes both these interfaces available to users. For routine monitoring of production status, programs were written to do a daily test of both interfaces. The programs and testing process will be presented.
- User reports gateway down.
- Alert on Failure or Confirm Good?
- Two interfaces?
Outline

1. Outline
   - Introduction
   - REST Interface
   - Browser Interface
curl or equivalent web client can be used to interact with NSG. This is explained in http://www.nsgportal.org/guide.html.

- Specify job parameters, including input file, and submit job
- Query job state
- Retrieve and verify output
Prerequisites

- User must have NSG account
- User must generate app key at https://nsgr.sdsc.edu:8443/restusers/login.action
Submit Job

```bash
curl  --config {mycurlconfigfile}
   -H cipres-appkey:{myappkey}
   https://nsgr.sdsc.edu:8443/cipresrest/v1/job/{myusername}
   -F tool=NEURON77_TG
   -F metadata.statusEmail=true
   -F vparam.number_cores_={mynumberofcores}
   -F vparam.number_nodes_={mynumberofnodes}
   -F vparam.runtime_={myruntime}
   -F vparam.filename_={myprogramfilename}
   -F 'vparam.cmdlineopts_={mycommandlineopts} ' 
   -F input.infile_=@{myzipfile}
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<jobstatus>
  <selfUri>
     DE4AD83D1040CE9109A0C3E9F081C4</url>
  </selfUri>
  <rel>jobstatus</rel>

...

Query Job Status

```
curl --config {mycurlconfigfile} 
   -H cipres-appkey:{myappkey} 
   {myselfUri}
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<jobstatus>
{ ... }

<jobHandle>NGBW-JOB-NEURON77_TG-29
    DE4AD83D1040CE9109A0C3E9F081C4</jobHandle>

<jobStage>SUBMITTED</jobStage>
{ ... }

This results in email with subject: ‘NSG-R successful job submit’
Store Job Info

- Job URL
- Expected output file
K. Yoshimoto

Gateway Production Monitoring
The next time the program runs, job status for all stored jobs will be checked. For COMPLETED jobs:

- Download the output.tar.gz file
- Extract the expected output file
- Compare output file with a reference output file, email successful or failed status
- Remove the work directory

NSG REST interactions are not difficult.
1 Outline

- Introduction
- REST Interface
- Browser Interface
Using NSG Portal

A web browser can be used to interact with NSG. This is explained in http://www.nsgportal.org/tutorial.html.

- Upload a zip archive for input. This should include a program and input data.
- Specify job parameters, including input data item, and submit job
- Query job state
- Retrieve and verify output
Things To Worry About

- Cookies (curl saves these)
- Token (must be parsed out of returned text)
- Data ID (must be parsed out of returned text by predetermined label)
- Task ID (must be parsed out of returned text by predetermined label)
Prerequisites

User must have NSG account
Simple Form

Sponsor email:
Applicant email:
Sponsor password:
Submit

Home

(Textfield "%s") Enter text. Use UP or DOWN arrows or tab to move off.

Enter text into the field by typing on the keyboard
Ctrl-U to delete all text in field, [Backspace] to delete a character
<html>
  <head>
  </head>
  <body>
    <form method="POST" action="/cgi-bin/sponsor.lsp">
      Sponsor email:<br>
      <input type="email" name="sponsoremail"/>
      <br>
      Applicant email:<br>
      <input type="email" name="applicantemail"/>
      <br>
      Sponsor password:<br>
    </form>
  </body>
</html>
<input type="password" name="sponsorpassword"><br>
<input type="submit" value="Submit"><br>
</form>
<p>
<a href="index.html">Home</a>
</p>
</body>
<html>
Welcome to NSG. Currently various neuroscience modeling and data processing tools can be run on High Performance Computing resources provided by the Extreme Science and Engineering Discovery Environment (XSEDE). Please note that if you are using human subject data, then the data should be de-identified. Please login to proceed.

First Time Users: Please review the usage policy.

New users who are interested in getting an account should fill out the form.

NSG Login:

Username

Password
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html;_charset=utf-8"/>
<!-- default value for selected tab -->
<link rel="icon" href="/portal2/images/favicon.ico" jsessionid=C15EA1628006822613DFBC55394F771A" />
<title>The Neuroscience Gateway | Login</title>
<link rel="stylesheet" href="/portal2/bootstrap-3.1.1-dist/css/bootstrap.min.css" jsessionid=C15EA1628006822613DFBC55394F771A" />
</head>

DOC TYPE html>
<link rel="stylesheet" href="/portal2/css/cipres.css;jsessionid=C15EA1628006822613DFBC55394F771A" type="text/css" media="screen, projection"/>
<!-[if IE 8]>
<link rel="stylesheet" href="/portal2/css/cipres-ie8.css;jsessionid=C15EA1628006822613DFBC55394F771A" type="text/css" media="screen, projection"/>
<![endif]->
<link rel="stylesheet" href="/portal2/css/messages.css;jsessionid=C15EA1628006822613DFBC55394F771A" type="text/css" media="screen, projection"/>
<link href="//netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.css" rel="stylesheet" />
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
<script type="text/javascript" src="/portal2/js/qtip/jquery.qtip.min.js;jsessionid=C15EA1628006822613DFBC55394F771A"></script>
<script type="text/javascript" src="/portal2/bootstrap-3.1.1-dist/js/bootstrap.min.js;jsessionid=C15EA1628006822613DFBC55394F771A"></script>
<script type="text/javascript" src="/portal2/js/site/core.js;jsessionid=C15EA1628006822613DFBC55394F771A"></script>
<script type="text/javascript" src="/portal2/js/ngbw.js;jsessionid=C15EA1628006822613DFBC55394F771A"></script>
<script type="text/javascript" src="/portal2/js/swami.js;jsessionid=C15EA1628006822613DFBC55394F771A"></script>
<script type="text/javascript" src="/portal2/js/checkboxgroup.js;jsessionid=C15EA1628006822613DFBC55394F771A"></script>
<script type="text/javascript" src="/portal2/js/switchcontent.js;jsessionid=C15EA1628006822613DFBC55394F771A"></script>
NSG Portal Login Page Source (cont.)

...
<form id="loginBox" name="loginBox"
    action="/portal2/login.action;
    jsessionid=C15EA1628006822613DFBC55394F771A"
    method="post" role="form">
  <div class="form-group">
    <label>Username</label>
    <input type="text" name="username"
          value="" id="loginBox_username"
          class="form-control" placeholder="Username"/>
  </div>
  <div class="form-group">
    <label>Password</label>
    <input type="password" name="password"
          value="" id="loginBox_password"
          class="form-control" placeholder="Password"/>
  </div>
</form>
<input type="password" name="currentPassword" size="Password" id="loginBox_currentPassword" class="form-control" />
</div>
<div class="form-group">
<input type="submit" value="Login" id="loginBox__login" name="method:login" class="btn btn-primary" />
</div>
Job Submit Details

- Log in, saving cookies
- Upload input zip archive, specifying a portal data item label that is specific for this run of the test
- Create a Task (Job), save the token associated with this task creation post
- Start data item selection process, using token from task creation
- Find data id by searching all data items for data item label
- Set data id to be used as input with created task
- Select Tool (application) to be used for this task
- Set parameters for this task
- Post request to save and run the task, using task label specific to this task
Get the task list, find the task id for the created task by the label we used earlier

Monitor the task until it goes into SUBMITTED state, then save the task info, with data label and task label
Output Verification Details

- Log in, save cookies
- For saved jobs, check status
- For COMPLETED jobs, retrieve output.tar.gz, compare output file to reference file
- Send appropriate success or failure email notification
- Delete working directory, uploaded data item, completed task (not quite working)
Summary

- Testing through the REST interface is straightforward.
- Testing through the browser interface is possible. This does not require a javascript engine for WF.
- Workbench Framework:
- kenneth@sdsc.edu
...<div class="callout">
    You can choose the following data.
</div>
<form id="createTask" name="selectData"
    action="/portal2/createTask.action"
    method="post">
    <input type="hidden" name="struts.token.name" value="token" />
    <input type="hidden" name="token" value="mytokenvalue" />
    <table class="table table-striped">
        <!-- Field Headers -->
    </table>
</form>...
<table>
<thead>
<tr>
<th>Data Item Label And ID</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>&lt;tr&gt;</strong></td>
</tr>
<tr>
<td><strong>&lt;td&gt;</strong></td>
</tr>
<tr>
<td><strong>&lt;input type=&quot;checkbox&quot; name=&quot;selectedIds&quot; value=&quot;43310&quot; id=&quot;createTask_selectedIds&quot; onclick=&quot;ids.check(this)&quot;/&gt;</strong>&lt;input type=&quot;hidden&quot; id=&quot;__checkbox_createTask_selectedIds&quot; name=&quot;__checkbox_selectedIds&quot; value=&quot;43310&quot; /&gt;**</td>
</tr>
<tr>
<td><strong>&lt;/td&gt;</strong></td>
</tr>
<tr>
<td><strong>&lt;td&gt;jones−20200306144630&lt;/td&gt;</strong></td>
</tr>
<tr>
<td><strong>&lt;td&gt;32855&lt;/td&gt;</strong></td>
</tr>
<tr>
<td><strong>&lt;td&gt;Unknown&lt;/td&gt;</strong></td>
</tr>
<tr>
<td>Date</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>3/6/20, 14:46</td>
</tr>
</tbody>
</table>

K. Yoshimoto
Gateway Production Monitoring
<a href="/portal2/task!display.action?id=51787" class="btn btn-link mc-replace" title="NGBW-JOB-NEURON77_TG-45041 F8F91274E4BA2D478207ACD14E9">jones-20200406160021</a>