OnlineHPC.com

Taverna web-based designer and High Performance Computing (HPC) platform
OnlineHPC is

• A two-year old project originated as a tool to simplify access to high performance computing (HPC) facilities for IITP researchers

• Web service for editing and running Taverna workflows available online at no cost

• Integrated with MyExperiment portal and VPH-Share portal to be available for a large community of Taverna users
Use cases

• Edit and execute workflows online

• Use cluster and cloud resources from within workflows or just to execute batch computing tasks

• Integrate your portal with online workflow editor

• Plug-in computational resources of your organization and custom Taverna plugins
Online Scientific Workflow Engine

OnlineHPC offers free-of-charge online scientific workflow editor. Sign up to try it! We also offer access to High Performance Computer cluster where you can perform your experiments.

Here are some screenshots of the editor.

Integrated High Performance Computing

Thousands of researchers know about cloud technologies and supercomputers and... continue time-consuming experiments with their desktops.

It is a gap between researcher skills and competence level needed to run high performance computing. There are at least three barriers on the way to HPC:

- Researcher needs to find HPC provider and go through procedures to get access;
- Researcher needs to install and configure numerous low-level software applications and deal with digital certificates to proceed;
- Researcher needs to get familiar with such technologies and tools as MPI, batch task managers or even web services.

The last requirement stops majority of even the stoutest researchers that passed first two levels.

We let simple things to be done in simple way:

- If you can run you experiment one time, nothing should stop you to run it 1000 times
- Combining multiple applications in a workflow is done with a powerful and nice online designer

OnlineHPC reduces barriers that stand on the way to HPC.
To run this example click the "Launch" button.

The Script component executes a script written in Java-like BEA WebLogic Jython. This example script will output the sum of two numbers given as "x" and "y" and put the result to the "sum" output parameter:

```
sum = Integer.parseInt(x) + Integer.parseInt(y)
```

You may add and remove ports to this component. Port names can be changed and variables in the script.

This is workflow input port.

This component contains constant "1000".
# Workflow export

<table>
<thead>
<tr>
<th>Title</th>
<th>Script example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Calculate the sum of two integers with a script component</td>
</tr>
<tr>
<td>External service</td>
<td>myExperiment</td>
</tr>
</tbody>
</table>
**Web interfaces for your Taverna components**
- Make components available for publicity without installation
- Provide complex services that include workflow components and access to backbone computing resources and databases
- Popularize your custom components and services

**Integration of your research and development environment with OnlineHPC**
- Let your users view, edit and even run workflows online
- Set up Single Sign-On between your enterprise portal and OnlineHPC
- Request dedicated OnlineHPC installation

**Provide simplified access to your HPC resources to your users or sell them to third party users**
- Let users run batch computing tasks or workflows that use clouds or clusters via a simple and user-friendly web app
- Combine your computing power and applications into a service that may be run alone or inside a Taverna workflow
Editable XML Structure

```xml
<mxGraphModel>
  <root>
    <mxCell id="0"/>
    <mxCell id="1" parent="0">
      <mxCell id="21" vertex="1" parent="1" idClass="d71a887e-a422-4d8b-af91-5001b4dde108" cellType="component">
        <mxComponentValue name="Script" idClass="d71a887e-a422-4d8b-af91-5001b4dde108" as="value">
          ...
          <mxRetry as="retry">
            <Object maxRetries="0" initialDelay="1000" maxDelay="5000" backoffFactor="1.0" as="properties"/>
          </mxRetry>
          <Object as="expand">
            <Object script="out=in;" as="beanshell"/>
          </Object>
        </mxComponentValue>
        <mxGeometry x="160" y="160" width="120" height="60" as="geometry"/>
      </mxCell>
      <mxCell id="22" vertex="1" parent="21" cellType="port" direction="in">
        <mxPortValue depth="0" name="in" as="value"/>
        ...
      </mxCell>
    </mxCell>
  </root>
</mxGraphModel>
```
JS component template

(function() {
    var config = {
        groupName : 'Processors',
        idClass : 'd71a887e-a422-4d8b-af91-5001b4dde108',
        name : 'Script',
        icon : 'icons48/gear.png',
        tuning : {
            'Advanced' : 'variable',
            inputSettings : { columns : ['name', 'depth', 'merge'] },
            outputSettings : { columns : ['name', 'depth'], direction : 'out' },
        }
    };

    com.onlinehpc.func.registerComponent(config, onNormalizeValue, onCreateComponent, onConfigureComponent);

    function onNormalizeValue(value) {...} // Validate user input and provide default values here
    function onCreateComponent(component) {...} // Add ports to graphical component
    function onConfigureComponent(descriptor) {...} // Create tabs and inputs
})();
Don’t hesitate to contact me

Vadim Surpin
vadim@iitp.ru

Thank you
&
Welcome to OnlineHPC