

Advanced Filter Profile

GetFactoryAttributesDocument

- Provides:
 - Factory Attributes (defined)
 - Activities in the container (open ended - jobs)
 - Resources represented in the container (large)
- BasicFilter: Scopes document size
 - Include activities: Y/N
 - Contained resources: Y/N
- Frequently need finer grained control

Advanced Filter - Activities

- Constrain the returned Activities by:
 - UserName: Running user id
 - Owner: Submitting user id
 - State: Activity state (Enumerated state)
 - ActivityIdRange: Start → End job id
 - DateTimeRange: From → Until date time
- Same terms are OR together
- Different terms are AND together
- An implementation MAY not support a filter term
 - But MUST then return `UnsupportedFeatureFault`
- An implementation MAY not return activities due to privacy concerns
 - But MUST then return `NotAuthorizedFault`

Example

- In GetFactoryAttributesDocument request

```
<hpcp-af:AdvancedFilter>
```

```
  <hpcp-af:UserName>Bob</hpcp-af:UserName>
```

```
  <hpcp-af:State>Pending</hpcp-af:State>
```

```
  <hpcp-af:State>Running</hpcp-af:State>
```

```
</hpcp-af:AdvancedFilter>
```

- Matching activities in existing response document
 - Activities for Bob that are Pending or Running

Advanced Filter - Resources

- Use a CompactResources representation (T/F)
 - In a cluster many nodes are the same
 - For identical nodes concatenate node name field
- Response

Existing Representation

```
<ContainedResource>  
  <ResourceName>Node01</ResourceName>  
  <CPUSpeed>2000</CPUSpeed>  
  <PhysicalMemory>2048</PhysicalMemory>  
</ContainedResource>  
<ContainedResource>  
  <ResourceName>Node02</ResourceName>  
  <CPUSpeed>2000</CPUSpeed>  
  <PhysicalMemory>2048</PhysicalMemory>  
</ContainedResource>
```

CompactResources Representation

```
<ContainedResource>  
  <ResourceName>Node01,Node02</ResourceName>  
  <CPUSpeed>2000</CPUSpeed>  
  <PhysicalMemory>2048</PhysicalMemory>  
</ContainedResource>
```

Advanced Filter - Resources

- Return the status of specific nodes
 - List of elements in NodeStatus element
- Response inserted in current response

```
<hpcp-af:DynamicNodeStatus>  
  <hpcp-af:NodeStatus name="ComputeNode01">  
    <hpcp-af:Memory used="512" free="1536" />  
    <hpcp-af:CPU used="3" free="5" load="18" />  
    <hpcp-af:Activity id="2304@enterprise.com" />  
    <hpcp-af:Activity id="2316@enterprise.com" />  
    <hpcp-af:Activity id="220@academic.org" />  
  </hpcp-af:NodeStatus>  
</hpcp-af:DynamicNodeStatus>
```

Application Template

Goal: Simplify job submission

- JSDL document references an application template
- Application template provided by service provider
- Template provides local values
 - Executable location
 - Queue parameters
 - Environment variables
 - ...
- Reduce errors due to badly specified documents

Discovery

- ApplicationsTemplates element in GetFactoryAttributesDocument
- Contains multiple JSDL Application elements
 - Application Name
 - Application Version
 - Description
- Template names not (necessarily) portable between HPCBP endpoints

Using an Application Template (AT)

- Provide a `jsdl:ApplicationName` element
 - Can be qualified with a `jsdl:ApplicationVersion` element
 - If the Application Name is not matched to a template then normal JSDL semantics used
- User may also specify elements:
 - `jsdl:JobIdentification`
 - `jsdl:DataStaging` elements
- `jsdl:Resources` elements overridden by the AT

Precedence within HPCApplication

Element	Replaced by AT value(s)	Added to AT value(s)	Generally Specified by AT
Executable	X		X
Working Directory	X		X
Username	X		X
Input/Output/Error	X		
Argument		X	
Environment	X	X	

JSDL returned from HPCBP is after the AT has been applied

Next Steps For Both Specifications

- Draft on GridForge
- Being discussed on telecoms
- Implementers:
 - Prototype within Microsoft HPC
 - ...