Auditing Service Manual

Introduction

The Auditing Service is a persistent web service designed to keep track of where the jobs in a user's workflows run on the TeraGrid and to keep track of how many SUs (service units) are consumed by those jobs and workflows. The Auditing Service is required to realize the following use cases:

1. The gateway administrator needs to know where the jobs in a user's workflow are running.
2. The gateway administrator has been contacted by a TeraGrid system administrator and asked to account for a job that is running under the community user account of that gateway. The TeraGrid system administrator gives the gateway administrator the local job id of the job and the gateway administrator needs to figure out which job (and hence which workflow and user) this local job id belongs to.
3. The gateway administrator needs to know how much of the community allocation is consumed by each user's workflows.

At a high level, the Auditing Service works as follows. First the Experiment Builder portlet in the LEAD Portal contacts the Auditing Service to register a workflow with it that the user is requesting to launch. The Auditing Service subscribes to notifications sent by this workflow. As the workflow progresses, the application services send auditing notifications that the Auditing Service receives, parses and stores the information contained within. Once the workflow terminates the Auditing Service cancels its notification subscription. Periodically, in a separate thread, the Auditing Service contacts TeraGrid auditing and accounting databases to update job information.

Web Service API

The Auditing Service implements the following web service methods.

- **AuditWorkflow**
  Registers a user's workflow with the Auditing Service. Called by the Experiment Builder portlet. Input message is a set of workflow attributes like the workflow instance id, the notification topic, the name of the workflow, etc. A sample input message:

  ```
  <AuditWorkflow>
  <workflow>
    <name>workflow_name</name>
    <userId>users_id</userId>
    <workflowId>workflow_id</workflowId>
    <notificationTopic>notif_topic</notificationTopic>
  </workflow>
  </AuditWorkflow>
  ```

  The output message is a simple boolean indicating whether the workflow was successfully registered or not. A sample output message:

  ```
  <AuditWorkflowResponse>
    <status>true</status>
  </AuditWorkflowResponse>
  ```

- **GetAccountingInfoByWorkflowId**
  A query method that retrieves accounting information for a workflow and its jobs matching the provided workflow id.

- **GetAccountingInfoByJobId**
A query method that retrieves accounting information for the job matching the provided job id. Also returns accounting information for the workflow that this job belongs to.

- **GetAccountingInfoByUserId**
  A query method that retrieves accounting information for the user matching the user id provided. In particular, this method returns aggregate information such as total SUs consumed and total number of workflows and jobs.

---

**Auditing Notification Schema**

The Auditing Service listens for special auditing type notification messages sent by the application services after starting a Globus GRAM job. These notifications are defined by the following schema:

```xml
<complexType name="ApplicationAuditType">
  <complexContent>
    <extension base="wft:BaseNotificationType">
      <sequence>
        <element name="name" type="string" />
        <element name="jobHandle" type="anyURI" />
        <element name="host" type="string" />
        <element name="queueName" type="string" minOccurs="0" />
        <element name="jobId" type="string" minOccurs="0" />
        <element name="distinguishedName" type="string" />
        <element name="projectId" type="string" minOccurs="0" />
        <element name="rsl" type="string" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The attributes of this notification type are defined as follows:

- **name** - simply the fully qualified name of the service, e.g., "http://www.extreme.indiana.edu/lead\WRF_Forecasting_Model"
- **jobHandle** - the grid job id, e.g., "ADAS_Analysis-Pub_Mon_Jun_04_12_24_32_EDT_2007_9105131180974387903"
- **host** - host to which the job was submitted, but this is actually a misnomer because what is actually stored is the GRAM jobmanager handle, e.g., "grid-hg.ncsa.teragrid.org/jobmanager-pbs"
- **queueName** - not currently set or used, but this would be the name of the queue that the job had been submitted to
- **jobId** - not currently set or used, but this would be the local job id of the GRAM job
- **distinguishedName** - the DN of the credential used to authenticate with the GRAM job manager when submitting the job, e.g., "/C=US/O=National Center for Supercomputing Applications/CN=LEAD Community User"
- **projectId** - not currently set or used, this would be the id of the project that the job will be charged to
- **rsl** - the RSL (Resource Specification Language) used to specify the GRAM job

In addition, since the auditing notification extends the BaseNotificationType, auditing notifications also contain attributes giving the id of the workflow, the workflow timestep, the workflow node id, etc.
Workflow Auditing Scenario

auditing service phases:

- registration of workflow
- collection of job info (grid job id)
- resolving of grid job ids to local job ids
- retrieval of accounting information from TG accounting DB

End to end trace of how the auditing service works.

Now we'll do an end to end trace of how the auditing service works. There are four distinctive phases of the workflow auditing process.

1. **Workflow Registration Phase.** The Experiment Builder registers the workflow with the Auditing Service prior to launching it.
2. **Job Info Harvesting Phase.** The application services send auditing notifications that the Auditing Service listens for and persists.
3. **Local Job ID Retrieval Phase.** The Auditing Service looks up the mapping of the grid job id to the local job id.
4. **Accounting Information Retrieval Phase.** The Auditing Service looks up the accounting information associated with a job based on its local job id.

In the Workflow Registration Phase, the Experiment Builder calls the AuditWorkflow method of the Auditing Service. The critical information passed is the user ID (i.e., the user's DN) of the user who is launching the job, the workflow instance id, and the notification topic. The user ID is critical to be able to map auditing and accounting information back to the portal user. The workflow instance id is critical because the auditing notification message reference the workflow instance id and hence this is the key to associating jobs to workflows. The notification topic is critical since this will be used to make a subscription to the Notification Broker for notifications related to this workflow.

In the Job Info Harvesting Phase, the Auditing Service passively listens for auditing notifications and upon receipt stores the information contained in them in its database.

In the Local Job ID Retrieval Phase, the Auditing Service contacts the GRAM auditing database of the Globus container to which that job had been submitted. This call is made via an OGSA-DAI client. This phase happens on a periodic (in the current configuration, hourly) basis. Once the local job id has been retrieved, the information for that job is updated in the database.

The Accounting Information Retrieval Phase is very similar to the Local Job ID Retrieval Phase. In fact, they are executed together. They are separated here because accounting information is usually not available until after a nightly update of local accounting information from the local TeraGrid sites to the TeraGrid central database, whereas the local job id is available in the GRAM auditing database immediately after the job completes. Hence, typically the Auditing Service first obtains a job's local job id and then several hours later it will retrieve the allocation usage for that job.