Nordic Testbed for Wide Area Computing And Data Handling

5/11/2005

THE LOGGER SERVICE

Functionality Description and Installation Manual[*]

A.Konstantinov

[*]Comments to: aleks@fys.uio.no

Contents

1 Introduction

The Logger Service is one of services implemented in HTTP(S,G) and SOAP Framework [1] and run in *httpsd* server. It provides simple frontend to underlying MySQL database to store and retrieve an information about jobs processed by ARC middleware [2].

Corresponding client is provided as part of ARC installed on cluster's frontend.

2 Logger Service

The Logger service is configured through configuration file of HTTPS/G server. It supports following service-specific commands.

- *acl_read [group [group [...]]]*
- *acl_write [group [group [...]]]*
- *acl_query [group [group [...]]]*

If client belongs to one of groups listed in *acl_read* command it is allowed to retrieve information about job records. Those clients are still restricted to an information about jobs which they are owners of. And those listed in *acl_write* can add new records. Usually those groups match against host credentials of a clusters' frontends. Groups in *acl_query* are allowed to retrieve information about *all* jobs.

Other supported commands are:

- *sqlcontact [user [password]]*
- *sqlcontactsource path*

They provide username and password used to access MySQL database. *sqlcontactsource* points to file which contains username and password in first line. It is better to use *sqlcontactsource* and make it readable only for user who starts the server.

The Logger accepts and stores in database following information:

Name	*Type*	*Description*
start	date and time	Time when job was started being processed
end	date and time	Time when job finished being processed
cluster	string	Subject of cluster's PKI certificate
user	string	Subject of job owner's certificate
id	string	Identifier of the job
name	string	Name of the job given by owner
failure	string	Reason of job's failure
lrms	string	LRMS which was used to execute job
queue	string	Queue name in LRMS
rsl	string	Job's decription as sent to cluster
ui	string	Hostname/ip of user's machine
usedcpu	number	Amount of CPU seconds used by job (not provived yes)
usedmem	number	Amount of memory bytes used by job (not provived yes)

3 Interface

Logger service supports two SOAP methods for adding and retrieveing records. It accepts information about processed jobs encoded in *UsageRecord* XML record as part of SOAP operation *add*. Coresponding WSDL is avaialble in Appendix B.

Usage record currently consists of following elements

ngjobid identifier of a job as used in ARC tools

usersn subject of user's certificate

cluster subject of cluster's certificate

description description of a job as passed to computing resource

projectname

jobname nae of a job specified by user

clienthost hostname or/and certificate from which user submited a job

requestedcputime amount of CPU time requested in job's description (minutes)

requestedmemory amount of memory requested in job's description

requesteddisk amount of disk space requested in job's description

submissiontime time when job reached computing resource

localuser name/id of a user used to execute job on computing resource

queue name of a queue on computing resource where aplicable

lrms type of batch system used at computing resource (aka LRMS)

localjobid id of a job in LRMS

lrmssubmissiontime time job was passed to LRMS

lrmsendtime time LRMS finished processing job

nodename host names of computers which were executing job

nodecount number of computers which were executing job

exitcode numerical code returned by job's man executable

failurestring textual description of fialure which happened during processing of a job (if any)

usedcputime CPU time consumed by job during executing

usedmemory maximal amount of memory used by job while executing

usedwalltime time it took for job to finish executing

useddisk amount of disk space taken by job while executing

status current state of job - kept for compatibility with GGF [4] Usage Record

endtime time computing resource finished processing job

downloadtime time it took to gather all input files requested for job

uploadtime time it took to distribute all output files produced by job

4 Logger Client

Logger service can be communicated using utility *logger* installed into ${NORDUGRID_LOCATION}/libexec by default. It allows to add new records into database and to retrieve information. First functionality is used by a grid-manager to pass inforamtion about processed jobs.

```
logger [-h] [-d level] [-u url] {control_dir ...|-q [-x] [-e element,element,...] [-o offset] [-s size] [
```

Here possible options are

-h show reminder,

-d set debug output to *level*,

-u URL of central logging server (mandatory if *-q* is given),

-q retrieve information, otherwise - report job to database,

-x print obtained information in XML,

-o from which record to start,

-s how many records to show,

-e specify which elements are desired in output.
Posible values are: *start,end,cluster,user,id,name,failure,lrms,queue,rsl,ui,usedcpu,usedmem*.

query is SQL conditional expression used to select jobs' records.

5 Installation

5.1 Install server

Just follow any of installation instructions available at http://www.nordugrid.org/papers.html to build (optionally) and install server part of NorduGrid ARC. If You choose 0.5 branch You will not need to do anything special. In case of 0.4 branch You will have to recompile ARC with –enable-experimental option of configure.

5.2 Configure MySQL

Logger service uses MySQL database. So it is essential to configure database properly. Currently You will either need file grid-manager/https/logger/create.sql or use content of Appendix A. It is important that You look into that file and modify first 3 lines (especially password). You can keep them as is, but at least make sure they fit Your intentions.

You have to feed that file to MySQL server. Preferably using *myslq* utility. This can look like *cat create.sql | mysql -p -u root* . ut You should check manual of *mysql* and configuration of the server before You type that.

If everything works MySQL server should have new database called *nglogger* and new user also called *nglogger* allowed to do everything inside that database. If You modified create.sql names can be different. Inside that database ther should be table called *jobs*. Do not change this name. There will also be table *files*, but it is not used yet.

That's it. You have database configured.

5.3 Configure server and service

Follow instruction in [1] to configure server. Do not forget to read [5] and add '=' if You use central configuration file.

Then follow instructions in section 2 to configure logger service. Make sure *sqlcontact/sqlcontactsource* contain right information.

5.4 Run server

Follow instruction in [1] to start server. Look into log file for line which looks like "Added service logger at /some/path". If You do not see it try to look through file for anything suspicious.

Appendix A

```
CREATE DATABASE IF NOT EXISTS nglogger;
USE nglogger;
CREATE TABLE IF NOT EXISTS jobs (
 num INT UNSIGNED AUTO_INCREMENT,
 start DATETIME NULL DEFAULT NULL,
 end DATETIME NULL DEFAULT NULL,
 cluster VARCHAR(255) NOT NULL DEFAULT "",
 user VARCHAR(255) NOT NULL DEFAULT "",
 id VARCHAR(32) NOT NULL DEFAULT "",
 name VARCHAR(255) NULL DEFAULT NULL,
 failure VARCHAR(255) NULL DEFAULT NULL,
 lrms VARCHAR(32) NULL DEFAULT NULL,
 queue VARCHAR(32) NULL DEFAULT NULL,
 rsl TEXT NULL DEFAULT NULL,
 ui VARCHAR(255) NULL DEFAULT NULL,
 usedcpu INT NULL DEFAULT NULL,
 usedmem INT NULL DEFAULT NULL,
 KEY num(num)
);
```

Appendix B

```
<definitions targetNamespace="http://www.nordugrid.org/ws/schemas/ARCLoggerV2"
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:nl2="http://www.nordugrid.org/ws/schemas/ARCLoggerV2"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
 xmlns="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types>
 <schema targetNamespace="http://www.nordugrid.org/ws/schemas/ARCLoggerV2"
  xmlns="http://www.w3.org/2001/XMLSchema">
```

```
<import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
<simpleType name="ResultCode">
 <restriction base="xsd:string">
  <enumeration value="NoError"/>
  <enumeration value="UndefinedError"/>
 </restriction>
</simpleType>
<complexType name="Result">
 <sequence>
   <element name="Code" type="nl2:ResultCode"/>
   <element name="Description" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
  </sequence>
</complexType>
<complexType name="UsageRecord">
 <sequence>
   <element name="ngjobid" type="xsd:string"/>
   <element name="usersn" type="xsd:string"/>
   <element name="cluster" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="description" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="projectname" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="jobname" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="clienthost" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="requestedcputime" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="requestedmemory" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="requesteddisk" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="submissiontime" type="xsd:dateTime" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="localuser" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="queue" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="lrms" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="localjobid" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="lrmssubmissiontime" type="xsd:dateTime" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="lrmsendtime" type="xsd:dateTime" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="nodename" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="nodecount" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="exitcode" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="failurestring" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="usedcputime" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="usedmemory" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="usedwalltime" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="useddisk" type="xsd:int" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="status" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="endtime" type="xsd:dateTime" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="downloadtime" type="xsd:dateTime" minOccurs="0" maxOccurs="1" nillable="true"/>
   <element name="uploadtime" type="xsd:dateTime" minOccurs="0" maxOccurs="1" nillable="true"/>
   <any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
 </sequence>
</complexType>
```

```
   <complexType name="addRequest">
    <sequence>
     <element name="job" type="nl2:UsageRecord" minOccurs="1" maxOccurs="unbounded"/>
    </sequence>
   </complexType>
   <complexType name="addResponse">
    <sequence>
     <element name="result" type="nl2:Result"/>
    </sequence>
   </complexType>
   <complexType name="getRequest">
    <sequence>
     <element name="query" type="xsd:string" minOccurs="0" maxOccurs="1" nillable="true"/>
     <element name="offset" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
     <element name="size" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    </sequence>
   </complexType>
   <complexType name="getResponse">
    <sequence>
     <element name="result" type="nl2:Result"/>
     <element name="job" type="nl2:UsageRecord" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
   </complexType>

  <element name="add" type="nl2:addRequest"/>
  <element name="addResponse" type="nl2:addResponse"/>
  <element name="get" type="nl2:getRequest"/>
  <element name="getResponse" type="nl2:getResponse"/>
 </schema>
</wsdl:types>
<wsdl:message name="addRequest">
 <wsdl:part name="addRequest" element="nl2:add"/>
</wsdl:message>
<wsdl:message name="addResponse">
 <wsdl:part name="addResponse" element="nl2:addResponse"/>
</wsdl:message>
<wsdl:message name="getRequest">
 <wsdl:part name="getRequest" element="nl2:get"/>
</wsdl:message>
<wsdl:message name="getResponse">
 <wsdl:part name="getResponse" element="nl2:getResponse"/>
</wsdl:message>
<wsdl:portType name="ARCLoggerV2PortType">
 <wsdl:operation name="add">
  <wsdl:documentation>Add new information into database</wsdl:documentation>
  <wsdl:input name="addRequest" message="nl2:addRequest"/>
  <wsdl:output name="addResponse" message="nl2:addResponse"/>
 </wsdl:operation>
```

```
 <wsdl:operation name="get">
  <wsdl:documentation>Query information in database</wsdl:documentation>
  <wsdl:input name="getRequest" message="nl2:getRequest"/>
  <wsdl:output name="getResponse" message="nl2:getResponse"/>
 </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ARCLoggerV2" type="nl2:ARCLoggerV2PortType">
 <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
 <wsdl:operation name="add">
  <soap:operation soapAction=""/>
  <wsdl:input name="addRequest">
   <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="addResponse">
   <soap:body use="literal"/>
  </wsdl:output>
 </wsdl:operation>
 <wsdl:operation name="get">
  <soap:operation soapAction=""/>
  <wsdl:input name="getRequest">
   <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="getResponse">
  <wsdl:output name="getResponse">
   <soap:body use="literal"/>
  </wsdl:output>
 </wsdl:operation>
</wsdl:binding>
<wsdl:service name="ARCLoggerV2">
 <wsdl:documentation>Database to keep and search information about jobs executed at NG-enabled GRID sites</wsdl:docum
 <wsdl:port name="ARCLoggerV2" binding="nl2:ARCLoggerV2">
  <soap:address location="http://localhost:80"/>
 </wsdl:port>
</wsdl:service>
</definitions>
```

References

[1] HTTP(S,G) and SOAP Server/Framework.

[2] NorduGrid middleware, the Advanced Resource Connector. http://www.nordugrid.org/middleware/

[3] The Globus Alliance. http://www.globus.org

[4] The Global Grid Forum. http://www.ggf.org

[5] Configuration and Authorisation of ARC (NorduGrid) Services.