NORDUGRID-TECH-15
9/9/2007

WS-Based ARC Clients

First Prototype

Markus Nordén*, Thomas Frågåt†

Abstract

This NorduGrid Technical document gives an introduction on how to use the prototype client tools developed within KnowARC.

*Markus.Norden@tsl.uu.se
†Thomas.Fragat@fys.uio.no

Contents

1 Introduction

This document describes a set of simple client tools for the new Web Service [**?**] (WS) based Advanced Resource Connector [**?**] (ARC) middleware. These tools are prototypes and are intended to be a proof-of-concept for the new ARC.

So far, the only service that is available for the new ARC is the ARC Resource-Coupled Execution Service [**?**] (A-REX). Therefore, this prototype release of WS clients consists of job handling tools with the following functionality:

- submission of jobs;
- querying the status of jobs; and
- termination of jobs.

The names of the command line tools begin with "ap", which shall be interpreted as "ARC Prototype" to distinguish them from the full-fledged production tools.

Support for staging of data would also have been desirable, but this is not yet fully implemented on the server side. Therefore it could not be included in this client prototype either. Thus, it is not possible to run any jobs that depend on local files on the computer on which the submitting client runs. Neither is it possible to retrieve any form of output (not even stdout/stderr) by using the new client.

Behind the command line tools, there is a minimalistic library which is documented in Appendix A. The library will serve as a basis for the development of a generalized ARClib. This will be used by the new client tools of the production releases of the next generation of ARC.

2 Configuration

The tools described here do not perform any brokering, but submit jobs to a specific A-REX service. Which A-REX service to use is specified in a configuration file. The configuration file also specifies how to set up the communication chain for the client. The location of the configuration file is specified by the environment variable `ARC_AREX_CONFIG`. If there is no such environment variable, the configuration file is assumed to be `arex_client.xml` in the current working directory. An example configuration file is shown in Appendix B.

3 Job Submission

The `apsub` command is used to submit jobs to an A-REX service.

Usage `apsub <JSDL-file> <JobID-file>`

Arguments

`<JSDL-file>` The name of a file that contains the job specification in JSDL format.

`<JobID-file>` The name of a file in which the Job ID will be stored. This file is created when the command is used. The reason for storing the Job ID in a file is its length (approximately 500 characters) which makes it unsuitable to use as a command line argument for the other tools.

Output A message that the job was submitted and in which file the Job ID is stored if the submission succeded. If the submission failed for some reason, an error message is displayed.

4 Job Status Query

The `apstat` command is used to query the status of a job that has been submitted to an A-REX service.

Usage `apstat <JobID-file>`

Arguments

`<JobID-file>` The name of a file in which the Job ID is stored. This file was created by `apsub` when the job was submitted.

Output The status of the job, e.g. "Pending/Accepted", or an error message if the status request failed for some reason.

5 Job Termination

The `apkill` command is used to terminate a job that has been submitted to an A-REX service.

Usage `apkill <JobID-file>`

Arguments

`<JobID-file>` The name of a file in which the Job ID is stored. This file was created by `apsub` when the job was submitted.

Output A message that the job was terminated, or an error message if the termination failed for some reason.

A The A-REX Client Library

Besides the three command line tools described earlier in this document, there is also a small library on which the command line tools are based. The library consists of two classes; the `AREXClient` class and the `AREXClientError` class.

A.1 The `AREXClient` class

This class is a client class for the A-REX service. It provides methods for three operations on an A-REX service:

- Job submission
- Job status query
- Job termination

A.1.1 Constructor

```
AREXClient(std::string configFile="")
```

This is the constructor for the AREXClient class. It creates an A-REX client that corresponds to a specific A-REX service, which is specified in a configuration file. The configuration file also specifies how to set up the communication chain for the client. The location of the configuration file can be provided as a parameter to the method. If no such parameter is given, the environment variable `ARC_AREX_CONFIG` is assumed to contain the location. If there is no such environment variable, the configuration file is assumed to be `arex_client.xml` in the current working directory.

Parameters

`configFile` The location of the configuration file.

Throws An `AREXClientError` object if an error occurs.

A.1.2 Job submission

```
std::string submit(std::istream& jsdl_file)
```

This method submits a job to the A-REX service corresponding to this client instance.

Parameters

`jsdl_file` An input stream from which the JSDL file for the job can be read.

Returns The Job ID of the the submitted job.

Throws An `AREXClientError` object if an error occurs.

A.1.3 Job status query

```
std::string stat(const std::string& jobid)
```

This method queries the A-REX service about the status of a job.

Parameters

`jobid` The Job ID of the job.

Returns The status of the job.

Throws An `AREXClientError` object if an error occurs.

A.1.4 Job termination

```
void kill(const std::string& jobid)
```

This method sends a request to the A-REX service to terminate a job.

Parameters

`jobid` The Job ID of the job.

Throws An `AREXClientError` object if an error occurs.

A.2 The `AREXClientError` class

This is an exception class that is used to handle runtime errors discovered in the AREXClient class.

Inherits This class inherits `std::runtime_error`

A.2.1 Constructor

```
AREXClientError(const std::string& what="")
```

This is the constructor of the AREXClientError class.

Parameters

`what` An explanation of the error.

```
<ArcConfig
  xmlns="http://www.nordugrid.org/schemas/ArcConfig/2007"
  xmlns:tcp="http://www.nordugrid.org/schemas/ArcMCCTCP/2007">
  <ModuleManager>
    <Path>/usr/lib/arc/</Path>
  </ModuleManager>
  <Plugins><Name>mcctcp</Name></Plugins>
  <Plugins><Name>mcctls</Name></Plugins>
  <Plugins><Name>mcchttp</Name></Plugins>
  <Plugins><Name>mccsoap</Name></Plugins>
  <Chain>
    <Component name='tcp.client' id='tcp'>
      <tcp:Connect>
        <tcp:Host>127.0.0.1</tcp:Host>
        <tcp:Port>60000</tcp:Port>
      </tcp:Connect>
    </Component>
    <Component name='tls.client' id='tls'>
      <next id='tcp'/>
    </Component>
    <Component name='http.client' id='http'>
      <next id='tls'/>
      <Method>POST</Method>
      <Endpoint></Endpoint>
    </Component>
    <Component name='soap.client' id='soap' entry='soap'>
      <next id='http'/>
    </Component>
  </Chain>
</ArcConfig>
```

Figure 1: An examle of an `arex_client.xml` configuration file.

B Configuration Example

Figure 1 shows an example configuration file, which is an XML file. All configuration information is enclosed within an `ArcConfig` element. That element also specifies which XML manespaces to use.

The `ModuleManager` element contains paths to directories where loadable plugins can be found. In this case, there is only one path – `/usr/lib/arc/` – but there may be many.

The `Plugins` elements specify which plugins to load – in this case message chain components (MCCs) for TCP, TLS, HTTP and SOAP.

The `Chain` element, finally, defines how to combine the MCCs that have been loaded into a communication chain. In this case, a TCP MCC that connects to port 6000 on the local host is connected to a TLS MCC, which is connected to a HTTP that uses the POST method, and finally a SOAP MCC is connected to the HTTP MCC.

The client tools will use the SOAP MCC as an entry point when sending messages. The messages will then propagate through the MCCs in the chain as specified in the configuration file.

References

[1] *Web Services Architecture*, W3C. [Online]. Available: http://www.w3c.org/TR/ws-arch/

[2] M. Ellert *et al.*, "Advanced Resource Connector middleware for lightweight computational Grids," *Future Generation Computer Systems*, vol. 23, pp. 219–240, 2007.

[3] A. Konstantinov, *The ARC Computational Job Management Module - A-REX: Description and Administrator's Manual*, The NorduGrid Collaboration, NORDUGRID-TECH-14.