GridARM: Java API for MDS and NWS

mumtaz.siddiqui@uibk.ac.at

August 9, 2004

Abstract

This document describes Java APIs for MDS and NWS and example usage.

1 Introduction

This package provides Java APIs for Globus MDS and NWS service. The package can be used to access different attributes associated with grid resources especially compute resources. The package is developed under the assumption that MDS2, MDS4 and/or NWS service is properly configured and running, and some parameters are supplied or available under default configuration. COG_INSTALL_PATH/lib/*.jar must be in the classpath along with JavaNWS.jar and gridarm_common.jar.

1.1 Important Classes

Following classes are important for discovering resources and resource attributes.

1.1.1 ResourceDiscoverer

This is an abstract class. It provides following methods to access resource information by using a driver for GIS.

- public public ResourceSpecificationSet discover(): discovers all registered resources along with all found attributes.
- public ResourceSpecificationSet discover(String name): discovers attributes of a single resource given as 'name'.
- public ResourceSpecificationSet discover(String[] attributes) Discovers all resources and returns only given attributes.
- public abstract ResourceSpecificationSet discover(ResourceRequest rq) this is abstract method and must be implemented by the driver class.
- public static ResourceDiscoverer getResourceDiscoverer(DiscovererContext context): This method is used to load appropriate driver for a GIS based on the name provided in the context.

1.1.2 DiscovererContext

Used to load a particular ResourceDiscoverer. Here are useful methods to set configuration parameters for resourcer discoverer.

- public void setName(String name): (required) sets name of the information service to be used for resource discovering. e.g. mds2, nws, mds4
- public void setPort(int port): (optional)i sets port for information service. e.g 2135 in case mds2

- public void setBaseName(String baseName): (optional) used to set base dn for Ldap in canse of mds2 service.
- public void setAddress(String address): sets host address. e.g. the name/address of MDS index server, or address nws NS host.
- public String getSchemaType(): (optional) sets schema type possible values in case of MDS2 are glue or default. It is not used for nws.
- public void setQueryType(String queryType): sets query type. eg RSL, XPATH or Basic. Right now only basic is used.

1.1.3 ResourceRequest

This class is used to manipulate a resource query. Constraints can be set using a setRelation or setter methods for attributes. e.g. request.setRelation(Constants.CPU_AVAULABLE, "4", Relation.MIN) is equivalent to the RSL relation (cpuAvailable ¿= 4)
Or request.setRelation(Constants.HOST_NAME, "xyz", Relation.EXACT) is equivalent to
request.setRelation(Constants.HOST_NAME, "xyz") or request.setHostName("xyz")
and RSL (HostName=xyz). Use HostRequest class for compute resource hosts. See class methods and Constants for possible available relations.

1.1.4 ResourceSpecification

The discover method of resource discoverer returns resource specification set which can be iterated to access/traverse specifications returned for all or a set of resources.
See class ResourceSpecification, HostSpecification

2 Grid Information Services

2.1 MDS2

Mds2 service can be used for both default or glue schema. The possible attributes which can be retrieved are listed in Constants class, or given in HostSpecification class as setter/getter methods. The following parameters should be supplied with DiscovererContext or configured as default.

- HOST_NAME: Name or address of MDS index server.
- PORT: port number of index service
- BASE_DN: Base domain name
- SCHEMA_TYPE: Possible values are default — glue. Defaults to default

2.1.1 Default configuration

The default configuration can be retrieved by using
DiscovererContext.defaultContext() static method. For HOST, PORT and BASE DN, the method tries to retrieve values from grid-info.conf fonfiguration file, which is present in the GLOBUS LOCATION/etc directory. For it to work, you should pass GLOBUS_LOCATION as system property using -D flag at java command line. The property name should be org.globus.wsrf.container.webroot as it is being used in GT4 (WSRF) implementation. If you cannot pass this option as command line, then you can create a symbolic link to GLOBUS_LOCATION as 'globus' in your home directory. If this is also not possible then simply copy grid-info.conf file in ~/.globus/etc

2.1.2 Usage

For example usage please see the class
org.askalon.grms.tools.AskGIS

Here is a sample usage of the API for MDS2 service.

```
DiscovererContext ctx = DiscovererContext.defaultContext();
if (dname != null) {
  ctx.setName(dname);
}
if (schema != null) {
  ctx.setSchemaType(schema);
}

if (host != null) {
  ctx.setAddress(host);
}

if (bdn != null) {
  ctx.setBaseName(bdn);
}

ResourceDiscoverer rd = ResourceDiscoverer.getResourceDiscoverer(ctx);
ResourceSpecificationSet rsset = rd.discover(request);

Iterator itr = rss.iterator();
while (itr.hasNext()) {
  HostSpecification host = (HostSpecification)itr.next();
  StringBuffer sb = new StringBuffer();
  sb.append("\nHost Specification:\n-------------------\n  Contact:");
  sb.append("\n\tName: ").append(host.getName());
  sb.append("\n\tType: ").append(host.getType());
  sb.append("\n\tLRM Type: ").append(host.getLRMType());
  sb.append("\n\tAccess Point: ").append(host.getAccessPoint());

  sb.append("\n  OS: ");
  sb.append("\n\tName: ").append(host.getOsName());
  sb.append("\n\tVersion: ").append(host.getOsVersion());
  sb.append("\n\tRelease: ").append(host.getOsRelease());

  sb.append("\n  CPU:");
  sb.append("\n\tCount: ").append(host.getCpuCount());
  sb.append("\n\tSpeed: ").append(host.getCpuSpeed());
  sb.append("\n\tAvailable: ").append(host.getCpuAvailable());
  sb.append("\n\tCurrent: ").append(host.getCpuCurrent());

  sb.append("\n  Platform:");
  sb.append("\n\tName: ").append(host.getPlatform());
  sb.append("\n\tArchitecture: ").append(host.getArchitechture());

  sb.append("\n  Filesystem:");
  sb.append("\n\tCount: ").append(host.getFSCount());
  sb.append("\n\tTotal MB: ").append(host.getFSTotal());
  sb.append("\n\tFree MB: ").append(host.getFSFree());

  Enumeration e = host.deviceNames();
  while (e.hasMoreElements()) {
    String dname = (String)e.nextElement();
    sb.append("\n  Device:").append(dname);
    sb.append(host.getDevice(dname));
  }

  System.out.println(sb.toString());
}
```

2.2 NWS

Currently nws service can be used to access cpuAvailable, cpuCurrent of registered hosts. The information can be queried for all registered hosts or for a subset of registered host. If the configured properly bandwidth and latency values can be retrieved between a registered host and other target system. Inorder to work, the following parameters should be supplied or configured as default;

- NWS_NAME_SERVER (Required) The host name or IP address of the NWS name server is required. It can be supplied during initialization of nws driver for resource discoverer as described in example usage.

- NWS_MEMORY_HOST (Optional) This is optional, parameter. It represents the nws memory server's name or address.

2.2.1 Default configuration

TBD

2.2.2 usage

The following example describe a simple usage for nws service. It takes "olperer.dos.uibk.ac.at" as nws nameserver and finds NWS driver by using ResourceDiscoverer.gerResourceDiscoverer(ctx) method.

```
DiscovererContext ctx = new DiscovererContext();
ctx.setAddress("olperer.dos.uibk.ac.at"); // NWS NS

ResourceDiscoverer nws = ResourceDiscoverer.gerResourceDiscoverer(ctx);
ResourceRequest request = new ResourceRequest();
request.setCpuAvailable(1.5, Relation.MIN);

ResourceSpecificationSet rss = nws.discover(request);
Iterator itr = rss.iterator();
while (itr.hasNex()) {
  HostSpecification hs = (HostSpecification)itr.next();
  System.out.println(hs.getCpuAvailable());
  System.out.println(hs.getCpuCurrent());
}
```

2.3 MDS4

Not implemented yet.

2.4 Todo

These APIs only work for basic information, needs testing and your feed back for improvement.
http://dps.uibk.ac.at/~mumtaz/GridARM-Common.pdf