Some New Features in iRODS 3.1

2012 User Group Meeting

Michael Wan

3.1 irods New features

- NETCDF/iRODS integration
  - Add NETCDF functionalities to iRODS
- Added compression capability to ibun and iphybun
  - using gzip, bzip2 and zip
  - Create compressed archive with -D option
    - e.g., ibun -c -D bzip2
  - extract compressed archive – automatic because the DataType is registered in iCAT.
  - Added a –add option to ibun
    - Add files to existing compressed/uncompressed tar file
- Added a –wlock to iput and a –rlock to iget and irepl
  - lock data objects while the operation is in progress
  - A new API rcDataObjLock for server-server locking

More irods 3.1New features (cont)

- Checksum computation for COMPOUND class objects by
  - Staging to cache and then compute the checksum of the staged objects.
- A new -K option for iphybun
  - compute and register checksum value for the bundled subfiles and bundle files.
- A new --purgec option for iput/iget/irepl
  - purge the cache copy of a compound resource group immediate after the operation is done.
- A new --empty option for irm to remove bundle files only when they are empty.
- Multiple federated IES servers can now run on the same host.
- A new --bundle option for ils to list subfiles in a bundle file.
- A new rule - acSetRescSchemeForRepl for selecting resource for replication.
- A new rule - acPostProcForRepl for post replication processing.

IRODS NETCDF implementation

- Wrap NETCDF APIs into iRODS APIs and micro-services
- NETCDF operations performed on iRODS servers for NETCDF data stored in iRODS.
- Initial implementation
  - Inquiry and subsetting functionalities
  - More is needed
- 7 new iRODS (client/server) APIs
  - Wrap 16 basic NETCDF APIs (libnetcdf) and one higher level libcf subsetting function (nccf_get_vara)
  - nc_create, nc_open, nc_close
  - nc_inq_varid, nc_inq_dimid, nc_inq_dim, nc_inq_var
    - Inquiry functions
  - nc_get_vars_text, nc_get_vars_uchar, nc_get_vars_string, nc_get_vars_int, nc_get_vars_uint, nc_get_vars_longlong, nc_get_vars_ulonglong, nc_get_vars_float, nc_get_vars_double
    - Subsetting functions, nc_get_vars_ class – most comprehensive.
  - nccf_get_vara
    - Higher level subsetting function of libcf for CF (climate and forcast) data
  - API test program can be found in lib/test/src/nctest.c

NETCDF micro-services

- 12 new NETCDF micro-services
  - Allow NETCDF workflows to be performed on the iRODS servers through the rule engine.
  - One for each of the 7 APIs
  - 5 micro-services for accessing data elements in the new data structures
  - An example workflow - clients/icommands/test/rules3.0/netcdfTest.r

NETCTF example workflow – netcdfTest.r

sfc_pres_temp.nc – netcdf file containing pressure and pressure in a 2-D grid (longitude and latitude)

```
netcdfTest () {
        if (msiNcOpen (*ncTestPath, "0", *ncid) == 0) {
            writeLine("stdout", "msiNcOpen success, ncid = *ncid");
       .....
# inq longitude
        if (msiNcInqId ("longitude", 1, *ncid, *londimid) == 0) {
            writeLine("stdout", "msiNcInqId success, londimid = *londimid");
        .....
        if (msiNcInqWithId (*londimid, 1, *ncid, *inqOut) == 0) {
            writeLine("stdout", "msiNcInqWithId londimid success");
            if (msiNcGetArrayLen (*inqOut, *lonArrayLen) == 0) {
                writeLine ("stdout", "lonArrayLen = *lonArrayLen");
       .....
# inq latitude
        if (msiNcInqId ("latitude", 1, *ncid, *latdimid) == 0) {
            writeLine("stdout", "msiNcInqId success, latdimid = *latdimid");
        .....

INPUT *ncTestPath="/wanZone/home/rods/netcdf/sfc_pres_temp.nc"
OUTPUT ruleExecOut,*tempVaraOut
```

NETCTF example workflow – netcdfTest.r (cont)

```
 # variables
if (msiNcInqId ("pressure", 0, *ncid, *pressvarid) == 0) {
         writeLine("stdout", "msiNcInqId success, pressvarid = *pressvarid");
      .....
      if (msiNcInqWithId (*pressvarid, 0, *ncid, *pressinqout) == 0) {
         writeLine("stdout", "msiNcInqWithId pressvarid success");
         if (msiNcGetNumDim (*pressinqout, *ndim) == 0) {
            writeLine("stdout", "pressinqout ndim = *ndim");
      .....
         if (msiNcGetDataType (*pressinqout, *pressDataType) == 0) {
            writeLine("stdout", "msiNcGetDataType success pressDataType = *pressDataType");
      .....
     if (msiNcGetVarsByType (*pressDataType, *ncid, *pressvarid, *ndim, "0%0", "3%5", "1%1", *getVarsOut) == 0)
{
        writeLine("stdout", "msiNcGetVarsByType pressvarid success");
        if (msiNcGetArrayLen (*getVarsOut, *pressArrayLen) == 0) {
           writeLine ("stdout", "pressArrayLen = *pressArrayLen");
          for(*I=0;*I<*pressArrayLen;*I=*I+1) {
             msiNcGetElementInArray (*getVarsOut, *I, *element);
             if (*pressDataType == 5) {
# float. writeLine cannot handle float yet.
                msiFloatToString (*element, *floatStr);
                writeLine("stdout", "pressure *I: *floatStr");
             } else {
                writeLine("stdout", "pressure *I: *element");
             }
          }
```