CENTER FOR SPATIAL INFORMATION SCIENCE AND SYSTEMS

GEORGE MASON UNIVERSITY

HDF-EOS Java Application Programming Interfaces

Genong (Eugene) Yu, Ph.D.
Research associate

Liping Di, Ph.D.
Professor & Director of CSISS

{gyu, ldi}@gmu.edu

November 28-30, 2006

HDF and HDF-EOS Workshop X,
Upper Marlboro, MD

Outline

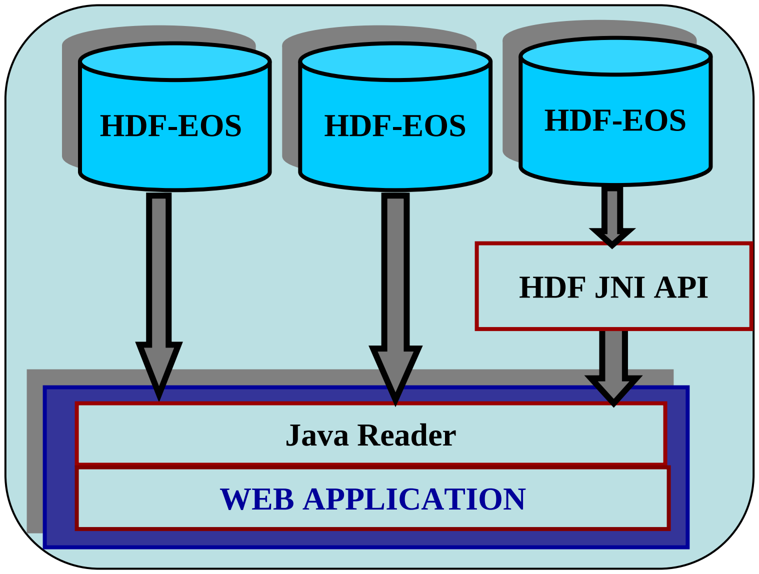
- Why Java API interfaces?
- Java API interfaces
  - Variable consideration
- Java Object wrap-up
- Performance consideration
- Applications
- Problem
  - Memory management

Demand

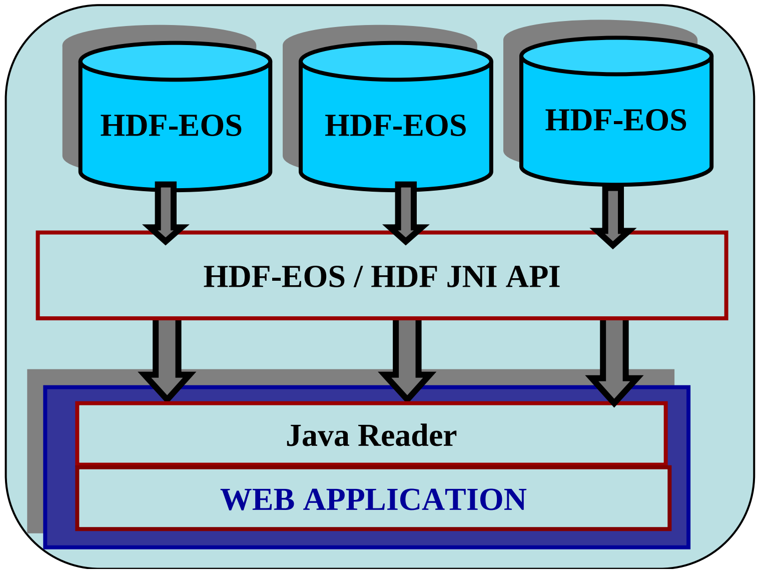
- Distribution systems over the Web
  - Java-based server
    - Apache Tomcat
- Manipulation of data for the Web application
  - Re-projection service
  - Classification service
  - Re-format service

Bridging Java Program and C/C++ library

The Problem



The Solution



- HDF-EOS library – in C/C++ library
  - All functions to manipulate grid and point are written in C
  - Primer for HDF-EOS is for C
  - HDF JNI libraries are available
- Possible approach
  - Rewrite the program in Java
    - Pros: better for Java environment
    - Cons: enormous work and need to keep track of revision
  - Call the library directly through JNI
    - Pros: manageable work and efficient re-use
    - Cons: memory management, debugging

How JNI works (1)

- Java interface
  - public static native int GDopen(String filename, int access) throws HDFEOSException;
  - Load library
    - System.loadLibrary("jhdfeos4");
- Compile the above code
  - javac HDFEOSLibrary.java
- Create header File
  - javah -jni edu

How JNI works (2)

- Create the library in C/C++
  - Template
    - JNIEXPORT void JNICALL Java_ClassName_MethodName (JNIEnv *env, jobject obj) {
    - //Implement Native Method Here
    - }
  - Example
    - JNIEXPORT jint JNICALL Java_edu_gmu_laits_hdfeos_HDFEOSLibrary_SWdupregion
    - (JNIEnv *env, jclass class, jint oldregionID)
    - {
    - int32 regionID;
    - regionID=SWdupregion((int32)oldregionID);
    - return (jint)regionID;
    - }

Tasks to enable the JNI

- To access a C library from a Java program, four tasks must be completed:
  - Declare appropriate “native” methods in Java classes
  - Write “interface” code for each native method to implement the JNI C or C++ standard
  - compile the JNI “interface” (C or C++) to create a JNI library
  - deploy the Java classes and the JNI library on the system

Mapping types (1)

- Interchangeable types

Native type (C/C++)	Java Language Type	Description	HDF/HDF-EOS
bool	jboolean	Unsigned 8 bits	intn, uint8
byte	jbyte	Signed 8 bits	int8
char	jchar	Unsigned 16 bits	uint16
short	jshort	Signed 16 bits	int16
long	jint	Signed 32 bits	int32
long long	jlong	Signed 64 bits	int64
Float	jfloat	Float 32 bits	float32
Double	jdouble	Float 64 bits	float64

Mapping types (2)

- String
  - Wrong
    - JNIEXPORT jint JNICALL Java_edu_gmu_laits_hdfeos_HDFEOSLibrary_SWcreate
    - (JNIEnv *env, jclass class, jint file_id, jstring swath_name)
    - {
    - return SWcreate((int32)file_id, (char *) swath_name);
    - }
  - Correct
    - JNIEXPORT jint JNICALL Java_edu_gmu_laits_hdfeos_HDFEOSLibrary_SWcreate
    - (JNIEnv *env, jclass class, jint file_id, jstring swath_name)
    - {
    - char *s_filename;
    - int32 swath_id;
    - **s_filename = (char *) (*env)->GetStringUTFChars(env, swath_name, 0);**
    - swath_id = SWcreate((int32)file_id, (char *)s_filename);
    - **(*env)->ReleaseStringUTFChars(env, swath_name, s_filename);**
    - return (jint)swath_id;
    - }

Mapping types (3)

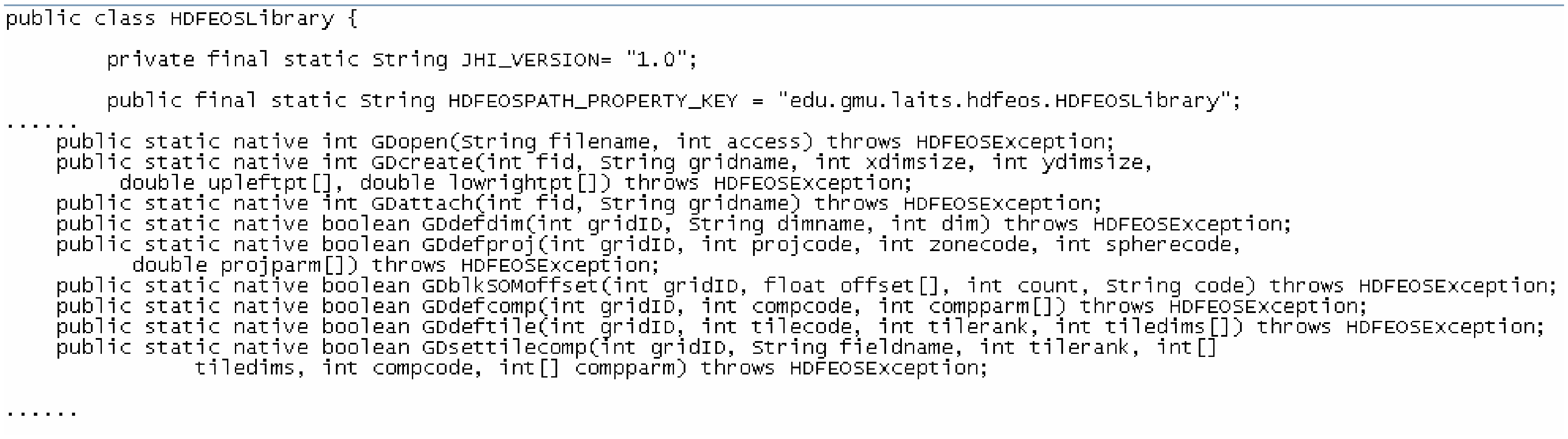
- Array
  - Wrong
    - JNIEXPORT jboolean JNICALL Java_edu_gmu_laits_hdfeos_HDFEOSLibrary_GDorigininfo
    - (JNIEnv *env, jclass class, jint gridID, jintArray origincode)
    - {
    - int32 status;
    - status= GDorigininfo((int32)gridID,(int32 *)origincode);
    - if (status==-1) return JNI_FALSE;
    - else return JNI_TRUE;
    - }
  - Correct
    - JNIEXPORT jboolean JNICALL Java_edu_gmu_laits_hdfeos_HDFEOSLibrary_GDorigininfo
    - (JNIEnv *env, jclass class, jint gridID, jintArray origincode)
    - {
    - int32 *i_origincode;
    - int32 status;
    - **i_origincode=(int32 *)(*env)->GetIntArrayElements(env,origincode,0);**
    - status=GDorigininfo((int32)gridID,i_origincode);
    - **(*env)->ReleaseIntArrayElements(env,origincode,(jint *)i_origincode,0);**
    - if (status==-1) return JNI_FALSE;
    - else return JNI_TRUE;
    - }

Mapping types (4)

- Pointer
  - In C
    - intn SWattrinfo(int32 swathID, char *attrname, int32 * numbertype, int32 *count);
  - In Java
    - public static native boolean SWattrinfo(int swathID, String attrname, int[] numbertype, int[] count) throws HDFEOSException;

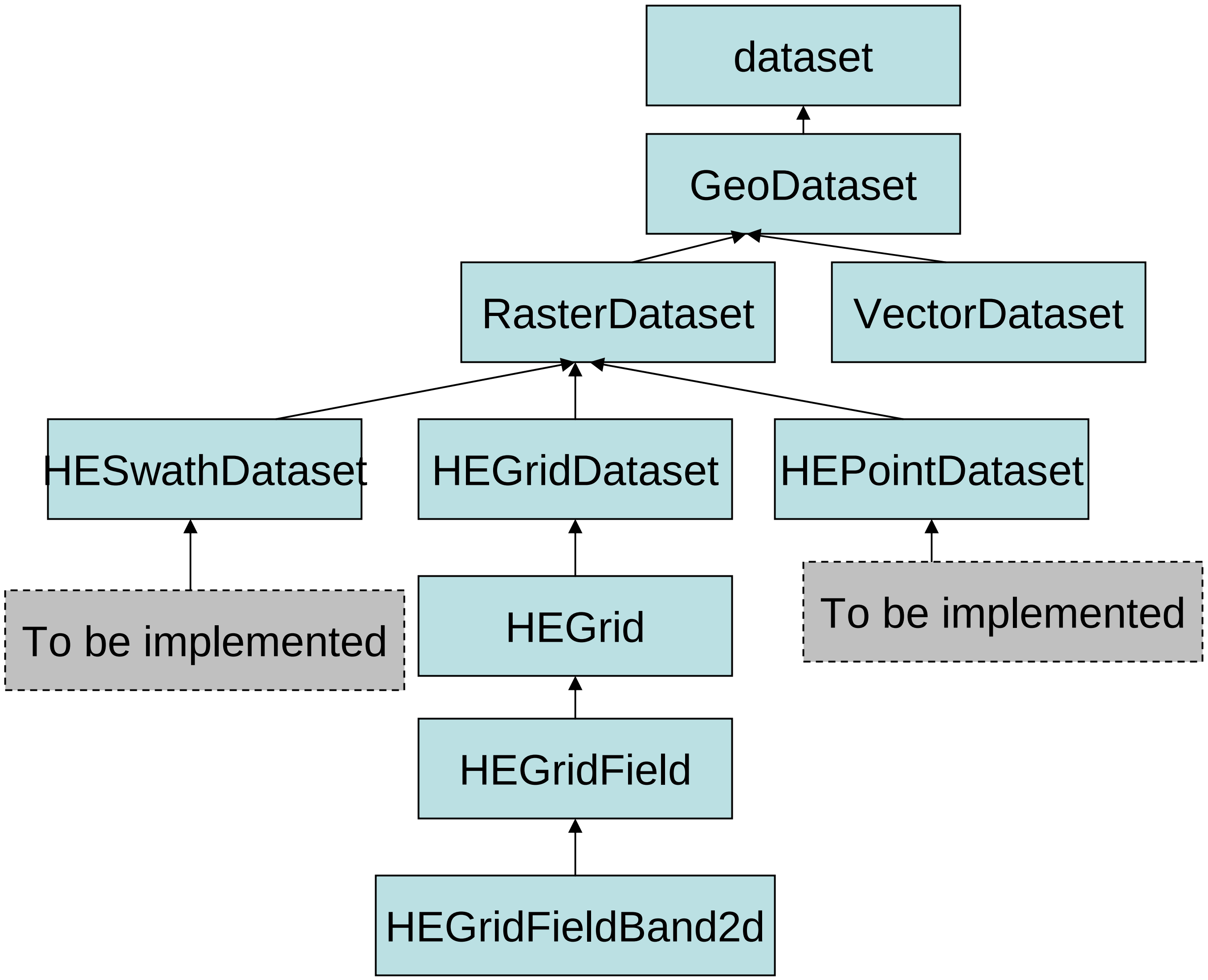
The HDF-EOS library

- Interface level
  - One class to hold all “native” methods
  - One C “interface” library – e.g. jhdfeos4 (dll, so)
  - One-to-one



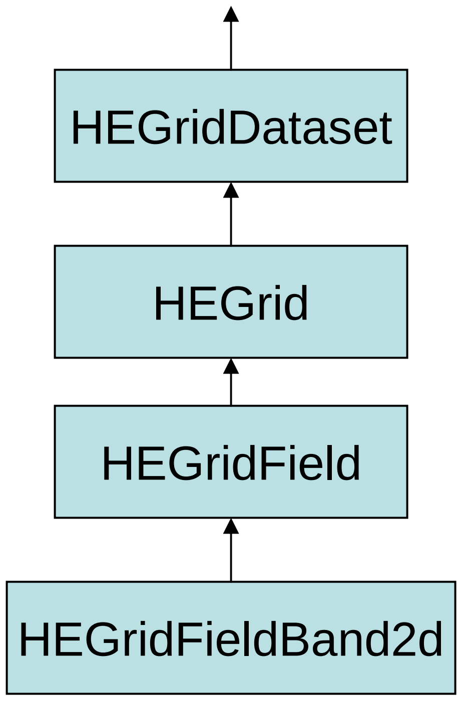
```
public class HDFEOSLibrary {

        private final static String JHI_VERSION= "1.0";

        public final static String HDFEOSPATH_PROPERTY_KEY = "edu.gmu.laits.hdfeos.HDFEOSLibrary";
......
    public static native int GDopen(String filename, int access) throws HDFEOSException;
    public static native int GDcreate(int fid, String gridname, int xdimsize, int ydimsize,
         double upleftpt[], double lowrightpt[]) throws HDFEOSException;
    public static native int GDattach(int fid, String gridname) throws HDFEOSException;
    public static native boolean GDdefdim(int gridID, String dimname, int dim) throws HDFEOSException;
    public static native boolean GDdefproj(int gridID, int projcode, int zonecode, int spherecode,
          double projparm[]) throws HDFEOSException;
    public static native boolean GDblkSOMoffset(int gridID, float offset[], int count, String code) throws HDFEOSException;
    public static native boolean GDdefcomp(int gridID, int compcode, int compparm[]) throws HDFEOSException;
    public static native boolean GDdeftile(int gridID, int tilecode, int tilerank, int tiledims[]) throws HDFEOSException;
    public static native boolean GDsettilecomp(int gridID, String fieldname, int tilerank, int[]
               tiledims, int compcode, int[] compparm) throws HDFEOSException;

......
```

Hierarchy of objects

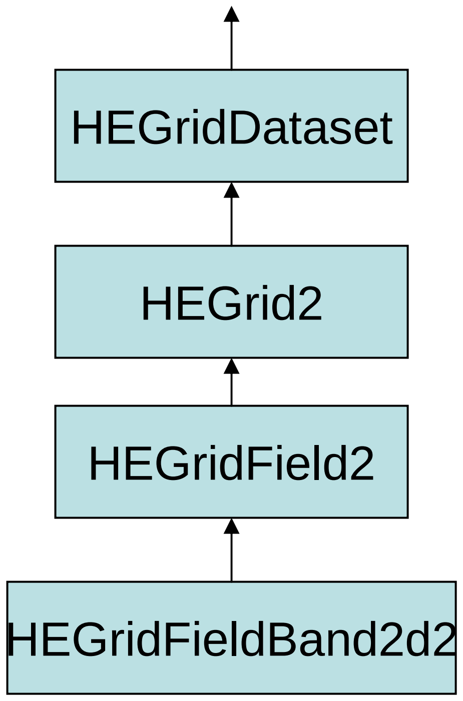


Thread safe vs. efficiency

- Model 1: Access & close
- Model 2: Open – access – close



Model 1

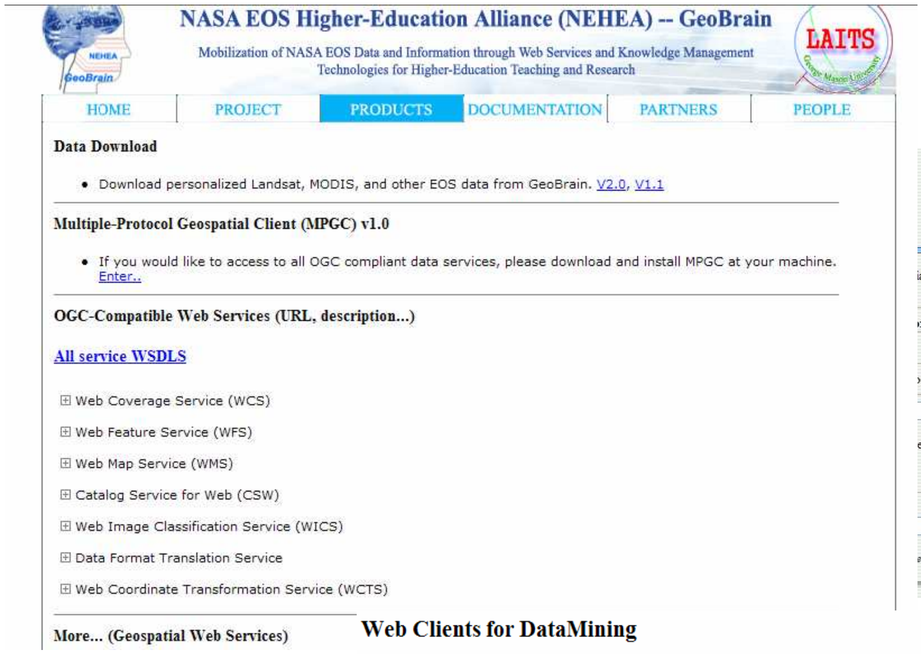


Model 2

Applications (1)

- Web services
  - Conversion http://laits.gmu.edu:18080/DataMining/HDFEOS2ARFF?WSDL
  - Conversion http://laits.gmu.edu:18080/DataMining/ARFF2HDFEOS2?WSDL
  - Training http://laits.gmu.edu:18080/DataMining/LogisticRegressionTrainer?WSDL
  - Classification http://laits.gmu.edu:18080/DataMining/LogisticRegressionClassifier?WSDL
  - Regression http://laits.gmu.edu:18080/DataMining/LogisticRegressor?WSDL
- Test pages http://laits.gmu.edu:18080/DataMiningClientWeb/

Applications (2)



NASA EOS Higher-Education Alliance (NEHEA) -- GeoBrain

Mobilization of NASA EOS Data and Information through Web Services and Knowledge Management Technologies for Higher-Education Teaching and Research

Data Download

- Download personalized Landsat, MODIS, and other EOS data from GeoBrain. V2.0, V1.1

Multiple-Protocol Geospatial Client (MPGC) v1.0

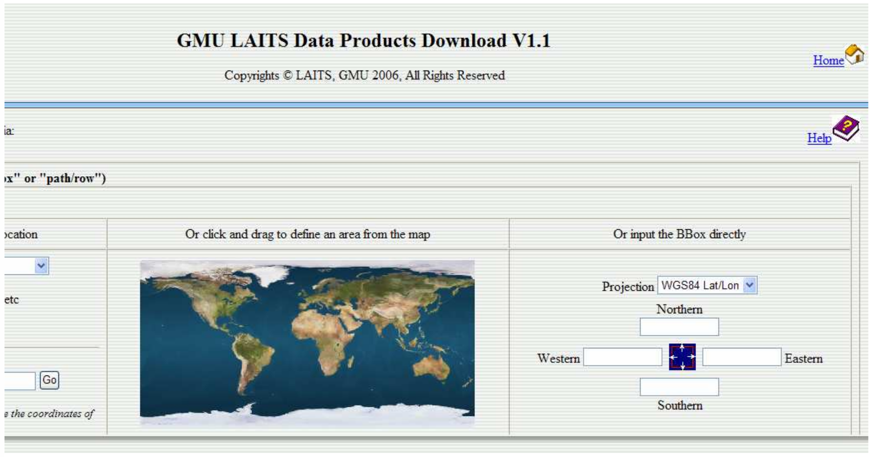
- If you would like to access to all OGC compliant data services, please download and install MPGC at your machine. Enter..

OGC-Compatible Web Services (URL, description...)

All service WSDLS

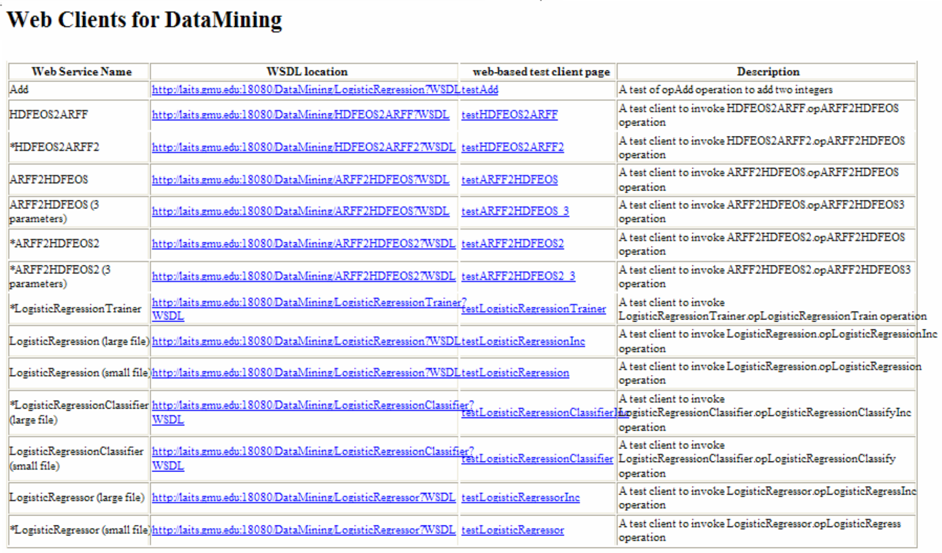
⊞ Web Coverage Service (WCS)
⊞ Web Feature Service (WFS)
⊞ Web Map Service (WMS)
⊞ Catalog Service for Web (CSW)
⊞ Web Image Classification Service (WICS)
⊞ Data Format Translation Service
⊞ Web Coordinate Transformation Service (WCTS)

More... (Geospatial Web Services)



GMU LAITS Data Products Download V1.1

Copyrights © LAITS, GMU 2006, All Rights Reserved



Web Clients for DataMining

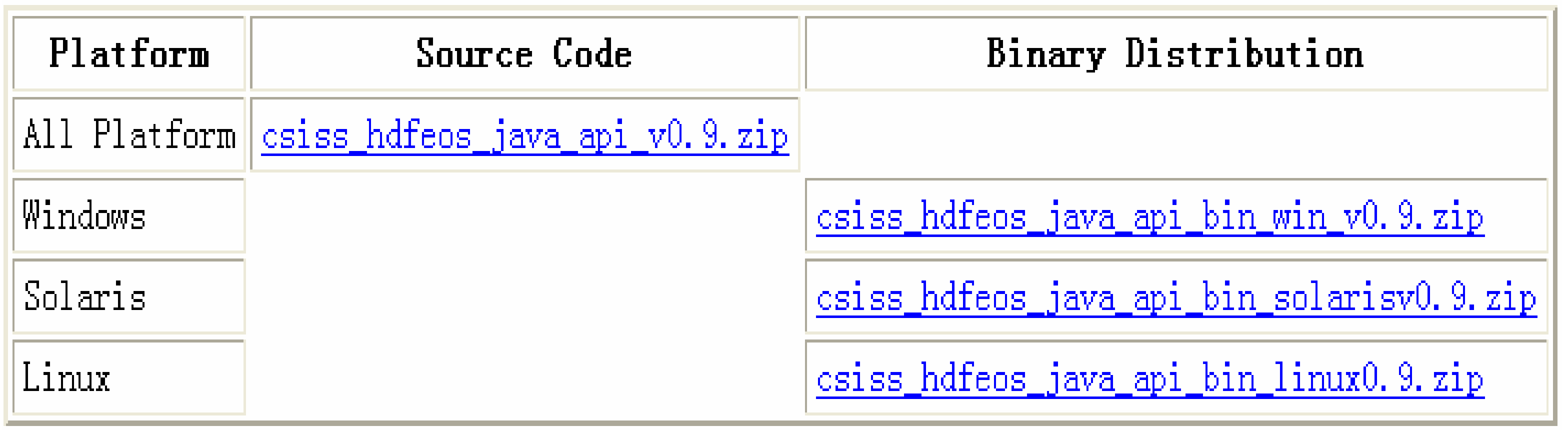
Web Service Name	WSDL location	web-based test client page	Description
Add	http://laits.gmu.edu:18080/DataMining/LogisticRegression?WSDL	testAdd	A test of opAdd operation to add two integers
HDFEOS2ARFF	http://laits.gmu.edu:18080/DataMining/HDFEOS2ARFF?WSDL	testHDFEOS2ARFF	A test client to invoke HDFEOS2ARFF.opARFF2HDFEOS operation
*HDFEOS2ARFF2	http://laits.gmu.edu:18080/DataMining/HDFEOS2ARFF2?WSDL	testHDFEOS2ARFF2	A test client to invoke HDFEOS2ARFF2.opARFF2HDFEOS operation
ARFF2HDFEOS	http://laits.gmu.edu:18080/DataMining/ARFF2HDFEOS?WSDL	testARFF2HDFEOS	A test client to invoke ARFF2HDFEOS.opARFF2HDFEOS operation
ARFF2HDFEOS (3 parameters)	http://laits.gmu.edu:18080/DataMining/ARFF2HDFEOS?WSDL	testARFF2HDFEOS_3	A test client to invoke ARFF2HDFEOS.opARFF2HDFEOS3 operation
*ARFF2HDFEOS2	http://laits.gmu.edu:18080/DataMining/ARFF2HDFEOS2?WSDL	testARFF2HDFEOS2	A test client to invoke ARFF2HDFEOS2.opARFF2HDFEOS operation
*ARFF2HDFEOS2 (3 parameters)	http://laits.gmu.edu:18080/DataMining/ARFF2HDFEOS2?WSDL	testARFF2HDFEOS2_3	A test client to invoke ARFF2HDFEOS2.opARFF2HDFEOS3 operation
*LogisticRegressionTrainer	http://laits.gmu.edu:18080/DataMining/LogisticRegressionTrainer?WSDL	testLogisticRegressionTrainer	A test client to invoke LogisticRegressionTrainer.opLogisticRegressionTrain operation
LogisticRegression (large file)	http://laits.gmu.edu:18080/DataMining/LogisticRegression?WSDL	testLogisticRegressionInc	A test client to invoke LogisticRegression.opLogisticRegressionInc operation
LogisticRegression (small file)	http://laits.gmu.edu:18080/DataMining/LogisticRegression?WSDL	testLogisticRegression	A test client to invoke LogisticRegression.opLogisticRegression operation
*LogisticRegressionClassifier (large file)	http://laits.gmu.edu:18080/DataMining/LogisticRegressionClassifier?WSDL	testLogisticRegressionClassifierInc	A test client to invoke LogisticRegressionClassifier.opLogisticRegressionClassifyInc operation
LogisticRegressionClassifier (small file)	http://laits.gmu.edu:18080/DataMining/LogisticRegressionClassifier?WSDL	testLogisticRegressionClassifier	A test client to invoke LogisticRegressionClassifier.opLogisticRegressionClassify operation
LogisticRegressor (large file)	http://laits.gmu.edu:18080/DataMining/LogisticRegressor?WSDL	testLogisticRegressorInc	A test client to invoke LogisticRegressor.opLogisticRegressInc operation
*LogisticRegressor (small file)	http://laits.gmu.edu:18080/DataMining/LogisticRegressor?WSDL	testLogisticRegressor	A test client to invoke LogisticRegressor.opLogisticRegress operation

Limitations to the JNI approach

- Limitations
  - JNI is not an easy API;
  - Invocation: only applications and signed applets can invoke the JNI;
  - Portability: No
    - compile different set of libraries and dynamically load
  - Memory management: no garbage collection
    - Careful to manage memory and exception in C
    - Timely re-start Tomcat server (not a good solution)
  - Debugging difficulty: error checking is a MUST or it has the potential
    - to crash the server
    - to left dead thread
    - to cause memory leak

Conclusions

- The library is available at
  - http://laits.gmu.edu/~gyu/HDFEOS/



Platform	Source Code	Binary Distribution
All Platform	csiss_hdfeos_java_api_v0.9.zip	
Windows		csiss_hdfeos_java_api_bin_win_v0.9.zip
Solaris		csiss_hdfeos_java_api_bin_solarisv0.9.zip
Linux		csiss_hdfeos_java_api_bin_linux0.9.zip

Future work

- Continue on updating the interface
- Work on HDF5-EOS
- Refine the object with considerations of performance and usability

References

- JNI specification http://java.sun.com/j2se/1.5.0/docs/guide/jni/spec/jniTOC.html
- Java Native Interface: Programmer’s Guide and Specification http://java.sun.com/docs/books/jni/

Acknowledgements

- The work was supported partially with grants from the NASA Earth Science Data and Information System Project (ESDISP) (NCC5-645, PI: Dr. Liping Di), NASA Earth Science Technology Office (ESTO) (NAG-13409, PI: Dr. Liping Di), NASA REASoN program (NNG04GE61A, PI: Dr. Liping Di), and National Geospatial-intelligence Agency (NGA) University Research Initiative (NURI) (HM1582-04-1-2021, PI: Dr. Liping Di).
- Thanks to Dr. Peisheng Zhao, Dr. Aijun Chen, Dr. Yuqi Bai, Mr. Yaxing Wei, and other colleagues at Center for Spatial Information Science and System, George Mason University, for their inputs and contributions.