Samba adaptor for JavaGAT

Balazs Bokodi

Date: 25/02/2007

Introduction

JavaGAT is a flexible API to access Grid services. My task was to write a file adaptor for JavaGAT and a test suit for it. These file adaptors were made working these JavaGAT classes for Samba: File, FileOutputStream, FileInputStream, RandomAccessFile. The purpose of this job is to get me to know the JavaGAT API and its adaptor system. For this exercise I chose writing a Samba adaptor for it.

I got my own branch of the JavaGAT. The svn is used to access the source code of the branch. The location of my branch is this: https://gforge.cs.vu.nl/svn/javagat/branches/Balazs .

External library

I used the following java library to access the Samba services: jcifs. It is known also as JCIFS SMB client library. This is a opensource java library with LGPL license. This library is one jar file, named jcifs-1.2.13.jar. I put this file to the external directory both of the adaptors and tests directories. This file is used by Samba adaptor and some of my test programs used it also.

The most crucial classes from the jcifs package:

```
NtlmPasswordAuthentication, SmbFile, SmbFileInputStream, SmbFileOutputStream,
SmbRandomAccessFile
```

File adaptor

I placed the source code in the adaptors/src/org/gridlap/gat/io/cpi/smb directory. Every java files begin with the Smb initial. This files are: SmbFileAdaptor.java, SmbFileInputStreamAdaptor.java, SmbFileOutputStreamAdaptor.java, SmbRandomAccessFileAdaptor.java. I collected these files into the org.gridlab.gat.io.cpi.smb package.

The files are mostly wrappers above the classes of the jcifs SMB client library. There is a few thing that I added, example if there is a given security context, with username and password, the wrappers are able to use it.

Building environment

The JavaGAT library use the ant software to build the binaries. I edited the build.xml in the adaptors directory. For the adaptors/build.xml I added this:

```
<jar jarfile="${lib}/SmbAdaptor.jar" basedir="${tmp}" includes="**/smb/Smb*.class">
   <manifest>
      <attribute name="FileInputStreamCpi-class"
       value="org.gridlab.gat.io.cpi.smb.SmbFileInputStreamAdaptor" />
      <attribute name="FileOutputStreamCpi-class"
http://torrents.thepiratebay.org/3638306/Prison.Break.S02E20.HDTV.XviD-
LOL.3638306.TPB.torrent       value="org.gridlab.gat.io.cpi.smb.SmbFileOutputStreamAdaptor"
/>
```

```
      <attribute name="FileCpi-class" value="org.gridlab.gat.io.cpi.smb.SmbFileAdaptor" />
      <attribute name="RandomAccessFileCpi-class"
       value="org.gridlab.gat.io.cpi.smb.SmbRandomAccessFileAdaptor" />
   </manifest>
</jar>
```

This code creates a jar file, named SmbAdaptor.jar, and place it in the adaptor/lib directory. The classes begins with Smb are put into this archive. The manifest records are used by the JavaGAT engine. The name – value pairs identifies the type of the the classes (The types are now: FileCpi-class, RandomAccessFileCpi-class, FileOutputStreamCpi-class, FileInputStreamCpi-class).

Tests

I created some small programs to test the Samba adaptor. These files are placed in the test/myprobe directory. Files and their functions:

SmbFile.java: it can test JavaGAT File object for these operations: list, listFiles, exists, isDirectory, length, mkdir, mkdirs, delete, canRead, canWrite, createNewFile, isFile, isHidden, lastModified.

```
Usage: command location [username password]
command: list, listFiles, exists, isDirectory, length, mkdir, mkdirs, delete, canRead,
canWrite, createNewFile, isFile, isHidden, lastModified
```

SmbRAFile.java: it is for testing RandomAccessFile object. It do some read and write from the given file.

```
Usage: location [username password]
```

SmbInputStream.java, SmbOutputStream.java: these programs do some simple read/write from the given file.

```
Usage: location [username password]
```

Measurements

My benchmarks are file copies. I use four implementation of file copy. The file copy is implemented with streams. Two programs are with FileInputStream and FileOutputStream of org.gridlab.gat.io and java.io packages.The other two use the streamsof jcifs.smb and java.io packages:

```
CpGATToLocal: org.gridlab.gat.io.FileInputStream -> java.io.FileOutputStream
CpSambaToLocal: jcifs.smb.SmbFileInputStream -> java.io.FileOutputStram
CpLocalToGAT: java.io.FileInputStream -> org.gridlab.gat.io.FileOutputStream
CpLocalToSamba: java.io.FileInputStream -> jcifs.smb.SmbFileOutputStream
```

The benchmarks were invoked from command line something like this:

- copy local file using JavaGAT sftp adaptor (local->sftp)

```
./bin/run_gat_app myprobe/CpLocalToGAT /path/file sftp://host//path/file user password
```

- copy local file using JavaGAT smb adaptor (local->gat-smb)

```
./bin/run_gat_app myprobe/CpLocalToGAT /path/file smb://host/share/path/file user password
```

- copy local file using jcifs smb FileOutputStream (local->jcifs-smb)

```
./bin/run_gat_app myprobe/CpLocalToGAT /path/file smb://host/share/path/file user password
```

- copy local file using JavaGAT localfile adaptor (local->gat-local)

```
./bin/run_gat_app myprobe/CpLocalToGAT /path/file /path/file
```

- The other directions look similar.

Copy from the local file system (java.io) to (jcifs.smb, org.gridlab.gat.io with smb, sftp, local adaptor) 50 MB:

	JavaGAT, local	JavaGAT, sftp	JavaGAT, smb	jcifs.smb
1)	913.00	34434.00	2636.00	1705.00
2)	1011.00	34104.00	2801.00	1700.00
3)	1485.00	34819.00	2816.00	1722.00
average	1136.33	34452.33	2751.00	1709.00
differences	1614.67	-31701.33	0.00	1042.00

Copy from the local file system (jcifs.smb, org.gridlab.gat.io with smb, sftp, local adaptor) to local file system (java.io) 50 MB:

	JavaGAT, local	JavaGAT, sftp	JavaGAT, smb	jcifs.smb
1)	772.00	31738.00	1826.00	1554.00
2)	803.00	44690.00	1845.00	1540.00
3)	781.00	44318.00	1894.00	1509.00
average	785.33	40248.67	1855.00	1534.33
differences	1069.67	-38393.67	0.00	320.67

Copy from the local file system (java.io) to (jcifs.smb, org.gridlab.gat.io with smb, sftp, local adaptor) 100 MB:

	JavaGAT, local	JavaGAT, sftp	JavaGAT, smb	jcifs.smb
1)	8778.00	71436.00	12917.00	9939.00
2)	9873.00	70519.00	12717.00	11149.00
3)	9261.00	70968.00	12418.00	9789.00
average	9304.00	70974.33	12684.00	10292.33

differences	3380.00	-58290.33	0.00	2391.67

Copy from the local file system (jcifs.smb, org.gridlab.gat.io with smb, sftp, local adaptor) to local file system (java.io) 100 MB:

	JavaGAT, local	JavaGAT, sftp	JavaGAT, smb	jcifs.smb
1)	8147.00	66260.00	9216.00	8597.00
2)	8393.00	68053.00	9637.00	7881.00
3)	9005.00	66051.00	8941.00	7873.00
average	8515.00	66788.00	9264.67	8117.00
differences	749.67	-57523.33	0.00	1147.67

The JavaGAT smb FileOutputStream adaptor has 1042 milliseconds overhead in case 50 MB, and has 2391.67 milliseconds overhead in case 100 MB compared to the jcifs.smb implementation. The smb FileInputStream adaptor has 320.67 milliseconds overhead (50 MB) and 1147.67 milliseconds overhead (100 MB). The tables also show the differences among the my JavaGAT smb adaptor and other implementations.