Grid, Storage and SRM

Alex Sim, Lawrence Berkeley National Laboratory, asim@lbl.gov
Jan. 30, 2008

1 Getting set up

Make a working directory for this exercise. For the rest of this exercise, all your work should be done in there.

```
$ mkdir srmex
$ cd srmex
```

Next set the environment variables to use for exercises. All exercises will be done on dCache SRM server at Fermi National Accelerator Laboratory (FNAL):

```
There are a few environmental variables already set for you.
      SRM_HOME : srm client installation directory; /sw/srmclient2
      SRMEP : SRM service endpoint; srm://gwdca04.fnal.gov:8443/srm/managerv2
      SRMPATH : Working directory on SRM storage; /pnfs/fnal.gov/data/osgedu
      MYNAME : your login
```

Next create a file to use for exercises:

```
$ dd if=/dev/zero of=smallfile-$MYNAME bs=1M count=2
$ ls -l
-rw-r--r-- 1 train03 train03 2097152 2008-01-12 18:15 smallfile-train03
```

1.1 Assumptions

You already have used **globus-url-copy** to move your files from your local machine to one of designated target machine and from a remote gridftp server to your local machine.

2 Basic operations

2.1 Checking the status of SRM

Use **srm-ping** to find out the status of SRM server on $SRMEP.

```
$ srm-ping $SRMEP
```

This returns SRM version number, similar to the following.

```
Ping versionInfo=v2.2
Extra information
    Key=backend_type
    Value=dCache
```

Key=backend_version
Value=production-1-8-0-9

2.2 Putting a file into SRM managed storage

File transfer into SRM managed storage goes through several protocols including gridftp file transfer. This client operation communicates with SRM server through several interfaces internally; srmPrepareToPut to request your file request, srmStatusOfPutRequest to check your request, gridftp file transfer and srmPutDone to finalize the state of your file transfer.

```
$ srm-copy file:////home/train99/srmex/smallfile-$MYNAME \
          $SRMEP\?SFN=$SRMPATH/smallfile-$MYNAME
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-CLIENT*REQUESTTYPE=put
SRM-CLIENT*TOTALFILES=1
SRM-CLIENT*TOTAL_SUCCESS=1
SRM-CLIENT*TOTAL_FAILED=0
SRM-CLIENT*REQUEST_TOKEN=-2146782625
SRM-CLIENT*REQUEST_STATUS=SRM_SUCCESS
SRM-CLIENT*SOURCEURL[0]= file:////home/train99/srmex/smallfile-$MYNAME
SRM-CLIENT*TARGETURL[0]= $SRMEP\?SFN=$SRMPATH/smallfile-$MYNAME
SRM-CLIENT*TRANSFERURL[0]=gsiftp://gwdca03.fnal.gov:2811///smallfile-alex
SRM-CLIENT*ACTUALSIZE[0]=2097152
SRM-CLIENT*FILE_STATUS[0]=SRM_SUCCESS
SRM-CLIENT*EXPLANATION[0]=Done
```

2.3 URL formats

A quick reminder on URL formats: We've seen two kinds of URLs so far.

- `file:////home/train99/srmex/smallfile` - a file called `smallfile` on the local file system, in directory `/home/train99/srmex/`. The appended `$MYNAME` is only to make the filename unique in this grid school.
- `srm://gwdca04.fnal.gov:8443/srm/managerv2\?SFN=/pnfs/fnal.gov/data/osgedu/smallfile-train99` – a SiteURL for a file name `smallfile-train99` on SRM running on the host called `gwdca04.fnal.gov` and port `8443` with the web service handle `/srm/managerv2` in directory `/pnfs/fnal.gov/data/osgedu`. SFN represents Site File Name.

2.4 Browsing a file in SRM managed storage

Now try to find out the properties of the file that you just put into SRM.

```
$ srm-ls $SRMEP\?SFN=$SRMPATH/smallfile-$MYNAME
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-CLIENT*REQUEST_STATUS=SRM_SUCCESS
SRM-CLIENT*REQUEST_EXPLANATION=srm-ls completed normally
SRM-CLIENT*SURL=/pnfs/fnal.gov/data/osgedu/smallfile-alex
```

```
SRM-CLIENT*BYTES=2097152
SRM-CLIENT*FILETYPE=FILE
SRM-CLIENT*STORAGETYPE=PERMANENT
SRM-CLIENT*FILE_STATUS=SRM_SUCCESS
SRM-CLIENT*OWNERPERMISSION=7166
SRM-CLIENT*LIFETIMELEFT=-1
SRM-CLIENT*LIFETIMEASSIGNED=-1
SRM-CLIENT*CHECKSUMTYPE=adler32
SRM-CLIENT*CHECKSUMVALUE=01e00001
SRM-CLIENT*FILELOCALITY=ONLINE
SRM-CLIENT*OWNERPERMISSION.USERID=7166
SRM-CLIENT*OWNERPERMISSION.MODE=RW
SRM-CLIENT*GROUPPERMISSION.GROUPID=9803
SRM-CLIENT*GROUPPERMISSION.MODE=R
SRM-CLIENT*OTHERPERMISSION=R
SRM-CLIENT*RETENTIONPOLICY=CUSTODIAL
SRM-CLIENT*ACCESSLATENCY=ONLINE
SRM-CLIENT*LASTACCESSED=2008-1-12-18-18-39
SRM-CLIENT*CREATEDATTIME=2008-1-12-18-18-39
```

2.5 Getting a file from SRM managed storage

Now try to get the file that you just browsed and put into SRM from the SRM managed storage to your local machine. This client operation communicates with SRM server through several interfaces internally: srmPrepareToGet to request your file request, srmStatusOfGetRequest to check your request, gridftp file transfer and srmReleaseFiles to release the file after your transfer.

```
$ srm-copy $SRMEP\?SFN=$SRMPATH/smallfile-$MYNAME \
           file:////home/train99/srmex/my-smallfile
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-CLIENT*REQUESTTYPE=get
SRM-CLIENT*TOTALFILES=1
SRM-CLIENT*TOTAL_SUCCESS=1
SRM-CLIENT*TOTAL_FAILED=0
SRM-CLIENT*REQUEST_TOKEN=-2146782626
SRM-CLIENT*REQUEST_STATUS=SRM_SUCCESS
SRM-CLIENT*SOURCEURL[0]= $SRMEP\?SFN=$SRMPATH/smallfile-$MYNAME
SRM-CLIENT*TARGETURL[0]= file:////home/train99/srmex/my-smallfile
SRM-CLIENT*TRANSFERURL[0]=gsiftp://gwdca03.fnal.gov:2811///smallfile-alex
SRM-CLIENT*ACTUALSIZE[0]=2097152
SRM-CLIENT*FILE_STATUS[0]=SRM_FILE_PINNED
SRM-CLIENT*EXPLANATION[0]=Done
```

After srm-copy is completed, find out the file size at the target on your local machine:

```
$ ls -l /home/train99/srmex/my-smallfile
-rw-r--r-- 1 train99 train99 2097152 2008-01-12 19:29 my-smallfile
```

2.6 Removing a file in SRM managed storage

Now try to remove the file that you put from the SRM managed storage.

```
$ srm-rm $SRMEP\?SFN=$SRMPATH/smallfile-$MYNAME
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-DIR: Total files to remove: 1
        status=SRM_SUCCESS
        explanation=successfully removed files
        surl=$SRMEP\?SFN=$SRMPATH/smallfile-$MYNAME
```

After `srm-rm` returns successfully, find out the file properties of the same SURL on the SRM with `srm-ls`. You should see that the SURL is invalid.

2.7 Creating and removing a directory in SRM managed storage

Now try to create a directory in SRM managed storage.

```
$ srm-mkdir $SRMEP\?SFN=$SRMPATH/$MYNAME
```

This will create a directory under the SRM that you can use in your SURLs.
Upon successful completion, this returns a summary similar to the following:

```
SRM-DIR: Sat Jan 12 19:04:09 CST 2008 Calling SrmMkdir
        status=SRM_SUCCESS
        explanation=success
```

Browse the directory to see what kind of property information that you retrieve from SRM.

Now try to remove the directory from SRM.

```
$ srm-rmdir $SRMEP\?SFN=$SRMPATH/$MYNAME
```

This will remove a directory under the SRM.
Upon successful completion, this returns a summary similar to the following:

```
SRM-DIR: Sat Jan 12 19:06:34 CST 2008 Calling SrmRmdir
        status=SRM_SUCCESS
        explanation=success
```

2.8 Summary of basic operations

Experiment with putting and getting files with different file sizes and numbers of parallel streams to and from the remote SRM site, and see the differences. When you use 4 parallel data streams by adding the `-parallelism` option with an argument of 4, the client operation goes through the same protocol, and the parallel streams are used in the gridftp file transfer. Larger files would make a significant difference in file transfer performance.

Experiment with directory structure in your path.

Note: Remember to remove those files and directories that you created afterwards.

3 Space management and related operations

3.1 Reserving a space in SRM for opportunistic use

Now, let's make a space reservation for 5M bytes of total space, 4M bytes of guaranteed space and lifetime of 900 seconds:

```
$ srm-sp-reserve –serviceurl $SRMEP –size 5000000 –gsize 4000000 –lifetime
900
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-SPACE: Status Code for spaceStatusRequest SRM_SUCCESS
        SpaceToken=258138
        TotalReservedSpaceSize=4000000
        Retention Policy=REPLICA
        Access Latency=ONLINE
```

Upon successful space reservation, this will show you the space token which will be used in the next exercises. (e.g. 258138 from above, but it is not necessarily numbers always and different storage may return different string format.) Note that your reserved space was returned as 4MB. Let's set the returned space token as an environment variable to re-use later on:

```
$ set SPTOKEN=258138
```

3.2 Finding out space properties from SRM

Now, let's find out the space information with the space token that you just received above:

```
$ srm-sp-info –serviceurl $SRMEP –spacetoken $SPTOKEN
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-SPACE:  ....space token details ....
        status=SRM_SUCCESS
        SpaceToken=258138
        TotalSize=4000000
        Owner=VoGroup=osgedu VoRole=null
        LifetimeAssigned=900
        LifetimeLeft=463
        UnusedSize=4000000
        GuaranteedSize=4000000
        RetentionPolicy=REPLICA
        AccessLatency=ONLINE
        status=SRM_SUCCESS
        explanation=ok
```

3.3 Retrieving space tokens from SRM

Supposed you lost your space token, and let's find out how to retrieve the space tokens that belong to you:

```
$ srm-sp-tokens –serviceurl $SRMEP
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-SPACE: ..................................
        Status=SRM_SUCCESS
        Explanation=OK
SRM-SPACE (0)SpaceToken=258138
```

This would show all the space tokens that belong to your grid identity and its mapping on the server.

3.4 Updating a space in SRM

Some time passed since the above space reservation, and the lifetime of the reserved space may be near the expiration. Now, let's update the lifetime of the space as well as the size of the space. We'll use 7MB of total space with 6MB of guaranteed space, and make the lifetime 950 seconds:

```
$ srm-sp-update –serviceurl $SRMEP –spacetoken $SPTOKEN –size 7000000 –gsize
6000000 –lifetime 950
```

Upon successful completion, this returns a summary similar to the following because the target SRM storage does not support this functionality.

```
SRM-SPACE: Sat Jan 12 19:09:55 CST 2008 Calling updateSpace request
        status=SRM_NOT_SUPPORTED
        explanation=can not find a handler, not implemented
        Request token=null
```

However, when the SRM storage supports the functionality and the request is successful, this returns a summary similar to the following.

```
SRM-SPACE: Sat Jan 12 21:22:50 PST 2008 Calling updateSpace request
        status=SRM_SUCCESS
        Request token=null
        lifetime=950
        Min=7000000
        Max=7000000
```

Your space token is the same as before, and upon successful completion, the lifetime and size of your space should be updated. Let's find out the space information from the SRM and verify using **`srm-sp-info`** to see the new updated information.

3.5 Putting a file into the reserved space in SRM

Now let's put a file into your reserved space using the space token. This client operation communicates with the SRM server, same as before. However, because of your space token, your file will be written into the space that you have reserved.

```
$ srm-copy file:////home/train99/srmex/smallfile-$MYNAME \
          $SRMEP\?SFN=$SRMPATH/smallfile-space-$MYNAME \
          -spacetoken $SPTOKEN
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-CLIENT*REQUESTTYPE=put
SRM-CLIENT*TOTALFILES=1
SRM-CLIENT*TOTAL_SUCCESS=1
SRM-CLIENT*TOTAL_FAILED=0
SRM-CLIENT*REQUEST_TOKEN=-2146782603
SRM-CLIENT*REQUEST_STATUS=SRM_SUCCESS
SRM-CLIENT*SOURCEURL[0]= file:////home/train99/srmex/smallfile-$MYNAME
SRM-CLIENT*TARGETURL[0]= $SRMEP\?SFN=$SRMPATH/smallfile-space-$MYNAME
SRM-CLIENT*TRANSFERURL[0]=gsiftp://gwdca03.fnal.gov:2811///smallfile-space-alex
SRM-CLIENT*ACTUALSIZE[0]=2097152
SRM-CLIENT*FILE_STATUS[0]=SRM_SUCCESS
SRM-CLIENT*EXPLANATION[0]=Done
```

After successful completion, find out the file properties with **srm-ls**.

```
$ srm-ls $SRMEP\?SFN=$SRMPATH/smallfile-space-$MYNAME
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-CLIENT*REQUEST_STATUS=SRM_SUCCESS
SRM-CLIENT*REQUEST_EXPLANATION=srm-ls completed normally
SRM-CLIENT*SURL=/pnfs/fnal.gov/data/osgedu/smallfile-space-alex
SRM-CLIENT*BYTES=2097152
SRM-CLIENT*FILETYPE=FILE
SRM-CLIENT*STORAGETYPE=PERMANENT
SRM-CLIENT*FILE_STATUS=SRM_SUCCESS
SRM-CLIENT*OWNERPERMISSION=7166
SRM-CLIENT*LIFETIMELEFT=-1
SRM-CLIENT*LIFETIMEASSIGNED=-1
SRM-CLIENT*CHECKSUMTYPE=adler32
SRM-CLIENT*CHECKSUMVALUE=01e00001
SRM-CLIENT*FILELOCALITY=ONLINE
SRM-CLIENT*OWNERPERMISSION.USERID=7166
SRM-CLIENT*OWNERPERMISSION.MODE=RW
SRM-CLIENT*GROUPPERMISSION.GROUPID=9803
SRM-CLIENT*GROUPPERMISSION.MODE=R
SRM-CLIENT*OTHERPERMISSION=R
SRM-CLIENT*SPACETOKENS(0)=258138
SRM-CLIENT*RETENTIONPOLICY=CUSTODIAL
SRM-CLIENT*ACCESSLATENCY=ONLINE
SRM-CLIENT*LASTACCESSED=2008-1-12-19-16-37
SRM-CLIENT*CREATEDATTIME=2008-1-12-19-16-37
```

Note from the previous srm-ls output that this time it shows the space token you used when putting your file into the SRM managed storage.

3.6 Releasing the reserved space from SRM

Now let's release the reserved space using the space token.

```
$ srm-sp-release –serviceurl $SRMEP -spacetoken $SPTOKEN
```

Upon successful completion, this returns a summary similar to the following:

```
SRM-SPACE: Releasing space for token=258138
        status=SRM_SUCCESS
        explanation=Space released
```

This operation may fail if you have any files in the space associated with the space token. In such case, remove the files with **srm-rm** to try releasing the space again.

```
$ srm-rm  $SRMEP\?SFN=$SRMPATH/smallfile-space-$MYNAME
```

After successful releasing your reserved space, find out the space properties with **srm-sp-info**.

3.7 Summary of space management operations

Experiment on reserving spaces with different space sizes and lifetimes, and putting your files into the reserved spaces with space token. Experiment updating the reserved space after you put your files into the reserved space. Experiment with directory structure in your SURL.

Note: Remember to remove those files and directories that you created afterwards. Also remember to release those spaces that you reserved if still active.

4 Advanced exercises (if time permits or as homework)

4.1 Setting up a small disk-based SRM

There are number of different SRM implementations, and for an advanced exercise, let's try to install and configure a small disk based SRM called BeStMan (Berkeley Storage Manager) for your personal use. Your personal SRM will be running on your grid proxy, and only you will be able to access it.

First, let's download the BeStMan, an SRM server, from the web: http://datagrid.lbl.gov/bestman. The current latest version is v2.2.0.6.tar.gz as of Jan. 2008. Untar the package: tar zxvf bestman-2.2.0.6.tar.gz. You will now have bestman/ directory in the current working directory.

As part of setup process, let's create a cache directory and a log directory in the current working directory: mkdir ./cache ./log
Find out the current working directory for full path information (normally $PWD corresponds to your current working directory).

Now, you will be using your grid proxy to launch your personal SRM, and find out where your grid proxy is located (it's normally located in /tmp/x509up_u####, and `id –u` would show your user id for #### part. let's call the proxy location as X509_USER_PROXY).

Next step is to decide which port number to assign for your personal SRM. In the example below, it shows two ports to use, 16249 and 16250. You can choose any two user ports for this exercise.
Also, the managed storage cache size is set up to 10MB, which is enough for your exercise.
Change your working directory to bestman/setup and let's configure.

```
./configure \
--with-replica-storage-path=$PWD/cache \
--with-replica-storage-size=10 \
--with-http-port=16249 \
--with-https-port=16250 \
--with-proxyfile-path=$X509_USER_PROXY \
--with-eventlog-path=$PWD /log \
--with-cachelog-path=$PWD /log
```

The configuration process creates new directories, and sbin is one of them in the bestman home directory. Change your working directory to sbin, and launch BeStMan SRM with "./SXXbestman.personal start". This starts up an SRM server for your own personal exercise that only you can access to.

To stop the server, change the directory to bestman/sbin, and type in "./SXXbestman .personal stop"

Now, you can go through the client commands that you have exercised, such as srm-ping, srm-copy, srm-ls, srm-rm, etc. and your own SRM endpoint from the example above is srm://`hostname`:16250/srm/v2/server, and your own directory path would be ~.
So, one example of an target URL on your SRM server can be srm://`hostname`:16250/srm/v2/server\?SFN=/srmcache/~/mysmallfile.

For an installation for a group or a site, please refer to https://twiki.grid.iu.edu/twiki/bin/view/Storage/BeStMan or http://www.dcache.org.

Have fun on your grid experiences, and when you have further questions on SRMs, you can send an email to srm@lbl.gov.