SymlinkDSpace

Contents

Symlink DSpace

Symlink DSpace is the project of the *Institute of Computer Science* at *Masaryk University* in Brno, Czech Republic. The main purpose of this project is to modify the original DSpace to work with big files (e.g. video files) in a more efficient way. The trick lies in using UN*X **symbolic links** on the file system layer.

Why Symlink DSpace?

The classic way the original DSpace works is "copying" files into the assetstore. All the files you want to have in DSpace have to be submitted via DSpace webUI or imported with help of *Import script*. From one point of view both ways works similarly. Submitting takes your locally stored file and transfers it to the DSpace server where the file is stored. Generally submitting is copying via network. Importing is copying naturally - files are copied from one directory to another.

Now assume we have a large amount of data stored on a local disk. And we want to have all these data stored in DSpace. Original DSpace *Import script* copies the data into the DSpace assetstore. If we have 1 TB of data then the procedure takes several hours 😞 . This is a wasting of disk performance and space...

Symlink DSpace avoids wasting disk performance and space because the files are not copied. There is only symbolic links to the files created (via "<tt>ln -s</tt>"

- so it is necessary to have the files on some "mounted" disk). No copying, no wasting. Because symlinks are done on the file system layer they are completely transparent to DSpace - for DSpace they are real files. Disadvantage is only that it is impossible to submit a file from a network connected computer (a file that is stored on "unmounted" disk).

Installation

All you have to do to install and use **Symlink DSpace** is to download the latest patch (for right version, now 1.3.2 only). In the <tt>$DSPACE_SRC</tt> directory run the following commands:

```
patch -Np1 < ../symlink-dspace-1.3.2.patch
```

Now choose the **Case 1:** or **Case 2:** to finish installation of **Symlink DSpace**.

Case 1: I have already installed DSpace

If you have already installed DSpace, then build and install only the necessary things with the following commands:

```
ant build_wars<br>
ant install_jar
```

Then copy the <tt>build/dspace.war</tt> to Tomcat's webapps directory and restart Tomcat (for other servers consult the manual).

Case 2: I am installing DSpace from scratch

If you want to install DSpace from scratch then it is easy. Just download and apply the patch (see above) and follow the classic installation steps (as described in the DSpace manual).

Usage

To do correct symlinking you can't submit/import the real files you want to have in DSpace!

Correct ingesting consists of two steps:

1. create a text file with content <tt> <the_path_to_file_for_ingest> </tt>
2. submit/import this text file

For example we want to ingest the file <tt> /disk1/data/video_file.avi </tt>:

1. create the file <tt> video_file.txt </tt> with content "<tt> /disk1/data/video_file.avi </tt>"
2. import the file <tt> video_file.txt </tt>

Now the file <tt> /disk1/data/video_file.avi </tt> is in DSpace (there is only a symlink in the assetstore). For details (why submitting text file) see **How does it work**.
Be careful about the user permissions! The user under which the DSpace is running has to be able to make symlinks to the file you want to ingest.
Symlink DSpace Import script provides a warning and writes a log file if you don't have enough permissions to make the symlink. But WebUI doesn't give you any warning! So if you submit the file via webUI and the file (bitstream) is not accessible probably you don't have the permissions to make symlink.

Some useful tips

It is not effective to submit to DSpace manually. We are using automatically generated (via **bash shell** scripts) structures that are then ingested through *Import script*. We can recommend using **Symlink DSpace** this way.

video@fi

<tt>video@fi</tt> is the project of *Faculty of Informatics* and *Institute of Computer Science* at *Masaryk University* in Brno, Czech Republic. The goal of the project is to store a large amount of video files. These video files are captured lessons that are taught at the Faculty of Informatics.

How does it work

Symlink DSpace is not a big project. The code for making symlinks is relatively simple. We found that rewriting the file ingest process in DSpace is not effective - it is not easy to rewrite the code and it is not comfortable to do rewriting with every new version of DSpace. It is more pleasant for the system (and for the developer too 🙂) to only add some code. This is the way we have chosen. Unfortunately it requires a small trade-off

- it is not allowed to give the files to DSpace directly. It is neccessary to use "a backdoor". The "backdoor" lies in submitting/importing only the text file that contents the path to the real file we want to make the symlink to.

For example:

- assume we want to submit/import file <tt> /disk1/data/files/video_file.avi </tt>
- we have to create text file <tt> video_file.txt </tt> with content "<tt> /disk1/data/files/video_file.avi </tt>"
- then we sumbit the text file <tt> video_file.txt </tt>

What does **Symlink DSpace** do? At first the text file with the path to the file is processed in the standard way (ID is generated, bitstream record is created and text file is stored in the assetstore). Then comes symlinking. The text file is opened and its content (a path to the real file) is read. Then the text file in assestore is replaced with a symbolic link to the real file. So the previous example continues with (all steps provides **Symlink DSpace**):

- open the submited <tt> video_file.txt </tt> and read its content "<tt> /disk1/data/files/video_file.avi </tt>"
- find the place in assetstore where <tt> video_file.txt </tt> is stored, assume the place is <tt>/dspace/assetstore/10/28/68/10286882464153</tt>
- execute the system calls:

```
rm -f /dspace/assetstore/10/28/68/10286882464153
ln -s /disk1/data/files/video_file.avi /dspace/assetstore/10/28/68/10286882464153
```

- update the DC metadata record:
  - name (video_file.txt -> video_file.avi)
  - size (e.g. 10k -> 600M)

And we are finished. Simple and powerful.

Contact, additional information

For any questions and suggestions please contact us at dspace@muni.cz. Project presentation that were presented at DSUG 2006 (http://dsug2006.uib.no/) in Bergen can be found http://library.muni.cz/symlinkdspace/ here.