

Video in DSpace

Vlastimil Krejčíř, krejcir@ics.muni.cz

Institute of Computer Science, Masaryk University, Brno, Czech Republic

DSpace User Group Meeting, Bergen 2006

Introduction

About our project

Project video@fi

The goal: capture almost all lessons taught at faculties at MU and make these video lessons accessible for students.

Players: Masaryk University and especially:

- Institute of Computer Science
- Faculty of Informatics

Started in autumn 2003.

Introduction II

Some basics facts

Present situation:

- over 1.5 TB of data
- more than 3000 files
- over 1500 captured lessons
- about 45 courses

Introduction II

Some basics facts

Present situation:

- over 1.5 TB of data
- more than 3000 files
- over 1500 captured lessons
- about 45 courses

And we want to store every captured lesson in **DSpace**.

Issues to solve

All files == a large amount of data.

DSpace submission proces consists of copying from one place to another (via network) – wasting time and disk performance.

And it breaks advantages of place where files are originally stored (breaks special caching, ...).

Especially copying is not nice :-(.

Issues to solve

All files == a large amount of data.

DSpace submission proces consists of copying from one place to another (via network) – wasting time and disk performance.

And it breaks advantages of place where files are originally stored (breaks special caching, ...).

Especially copying is not nice :-).

Solution: We can use UN*X **symbolic links!**

Symlinking I.

Issue: How to send (web UI) a complete path (/disk1/dir/...) to file we want to make a symlink to?

- Classic HTTP through `<input type="file" ... />` doesn't work :-(.
• There's no other `<input type="???" ... />`

Symlinking I.

Issue: How to send (web UI) a complete path (/disk1/dir/...) to file we want to make a symlink to?

- Classic HTTP through `<input type="file" ... />` doesn't work :-(.
 - There's no other `<input type="???" ... />`

Solution: Use Javascript... :-/

Symlinking II.

OK we send a path instead of a file => we need to completely rewrite inner submit system (including all appropriate classes).

We need to do this with every new version of DSpace :-).

We are lazy ;-). We need something better.

Our solution

Why not let DSpace do all the work? Let's do a dirty trick :-).

Submit the text file simply containing the complete path to the file!

DSpace processes it (with all necessary steps we need).

Our solution

Why not let DSpace do all the work? Let's do a dirty trick :-).

Submit the text file simply containing the complete path to the file!

DSpace processes it (with all necessary steps we need).

Then

- **read the path from the text file we submitted**

Our solution

Why not let DSpace do all the work? Let's do a dirty trick :-).

Submit the text file simply containing the complete path to the file!

DSpace processes it (with all necessary steps we need).

Then

- **read the path from the text file we submitted**
- **replace the file in the assetstore with the symlink**

Our solution

Why not let DSpace do all the work? Let's do a dirty trick :-).

Submit the text file simply containing the complete path to the file!

DSpace processes it (with all necessary steps we need).

Then

- **read the path from the text file we submitted**
- **replace the file in the assetstore with the symlink**
- **correct appropriate metadata**

and we are finished :-).

Summary

Advantages of our "dirty" trick:

- no code rewriting, only add simple one;
- no web UI change;
- no breaking of caches and other performance improvements.

Disadvantages:

- can submit only files which are stored at disks accessible from computer the DSpace runs (disks "mounted" to server) :-)

Disadvantages don't hurt us in this case :-).

Example

A file we want submit:

/disk1/videofiles/pv002_2006-11-23.avi

A text file containing the complete path above:

pv002_2006-11-23.txt

Submit pv002_2006-11-23.txt to DSpace. This text file is stored at assetstore (assume at \$STORED_TEXT_FILE). We have to do:

- replace \$STORED_TEXT_FILE with a symlink to /disk1/videofiles/pv002_2006-11-23.avi;
- correct the metadata (bitstream size, name and format)

Questions?

Thank you for your attention