TECHNISCHE UNIVERSITÄT DRESDEN

Media Center

TU Dresden: A Large-Scale Plone Deployment Case Study

Dresden, 10/20/17

DRESDEN concept
Exzellenz aus Wissenschaft und Kultur

Motivation

There is no real new stuff here

Provide feedback to the wider Plone community

Starting Point

Starting Point (2013)

Plone2.1

No Archetypes

Multilingual fields

No responsive design

Lots of technical debt

The Goal

Complete Relaunch of the university's website

Based on the latest and greatest Plone (at the time Plone 4.2)

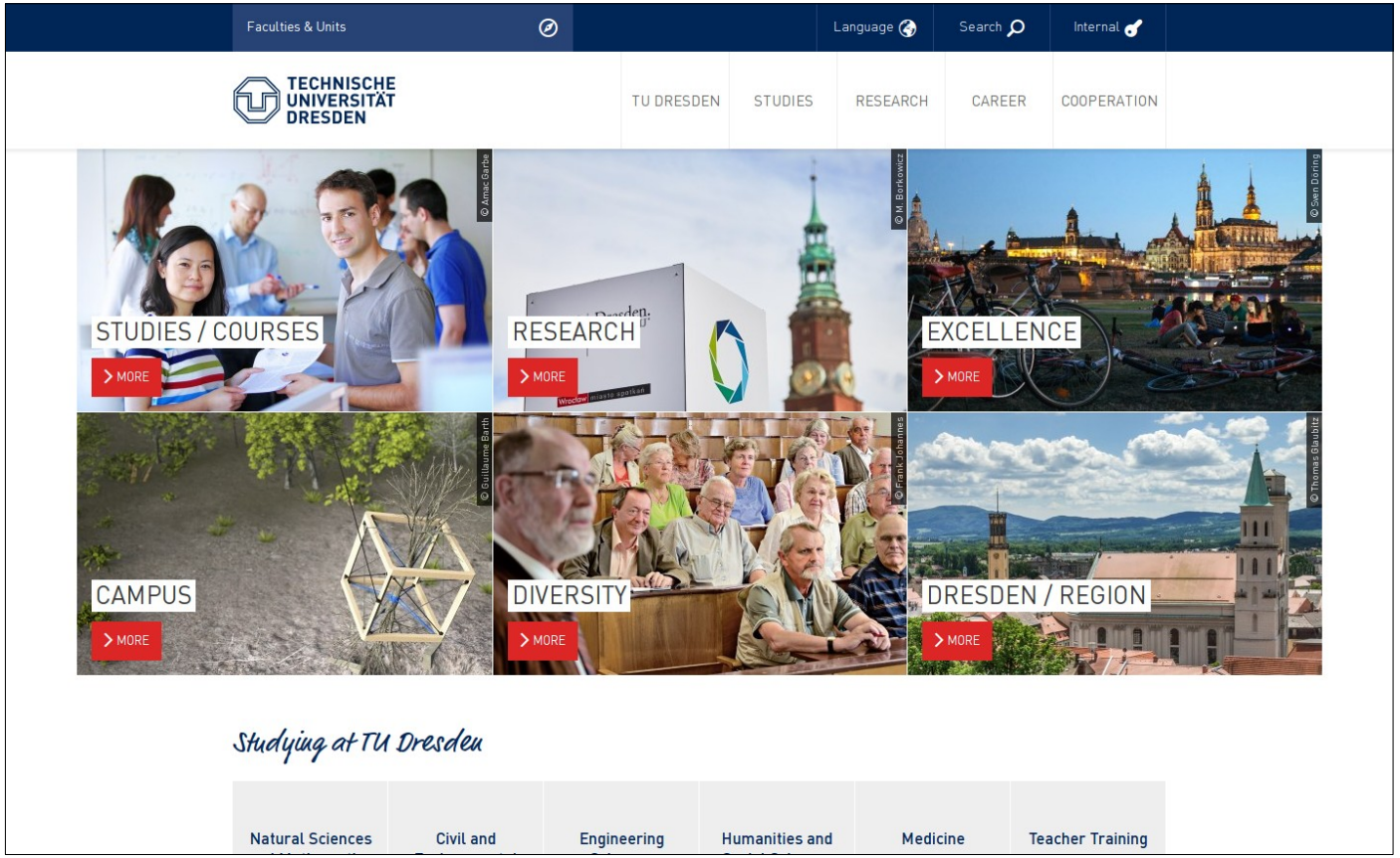
New responsive design

New site structure

Timeline

Jan 2013	First concept meeting
Jul 2013	Actual work on new TUD's WebCMS starts
Feb 2014	External agency delivers design prototype
Nov 2015	Content migration for central web pages
Mar 2016	Launch of the new TUD website
Until Feb 2017	Content migration for faculties, etc.
Sep 2017	Old WebCMS decommissioned

Result


Faculties & Units
Language
Search
Internal
TECHNISCHE UNIVERSITÄT DRESDEN
TU DRESDEN
STUDIES
RESEARCH
CAREER
COOPERATION
STUDIES / COURSES
MORE
RESEARCH
MORE
EXCELLENCE
MORE
CAMPUS
MORE
DIVERSITY
MORE
DRESDEN / REGION
MORE
Studying at TU Dresden
Natural Sciences
Civil and
Engineering
Humanities and
Medicine
Teacher Training

Result

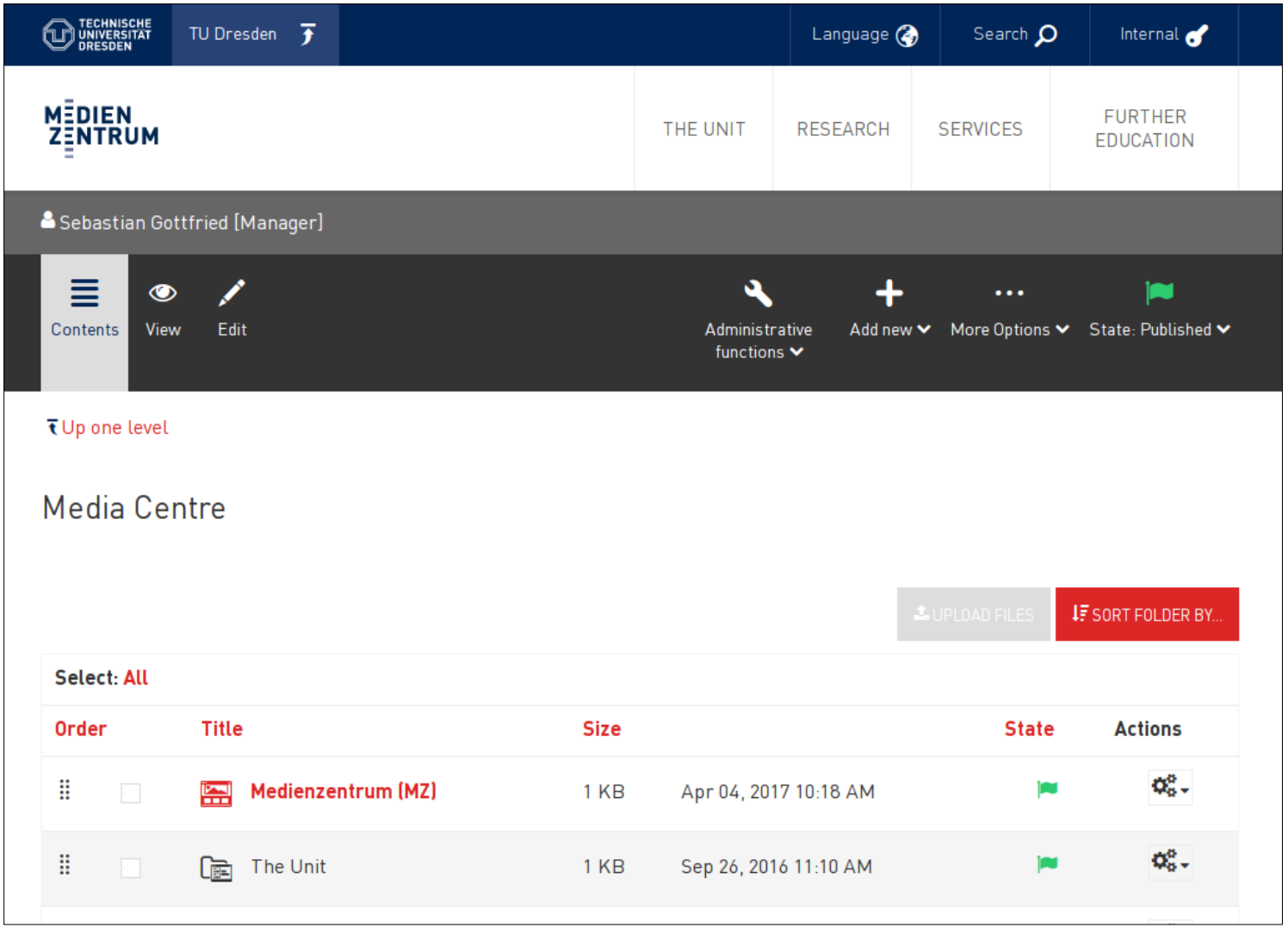
Plone 4.3.14

Archetypes

Multilingual fields with `raptus.multilanguagefields`

Heavily customized Plone UI

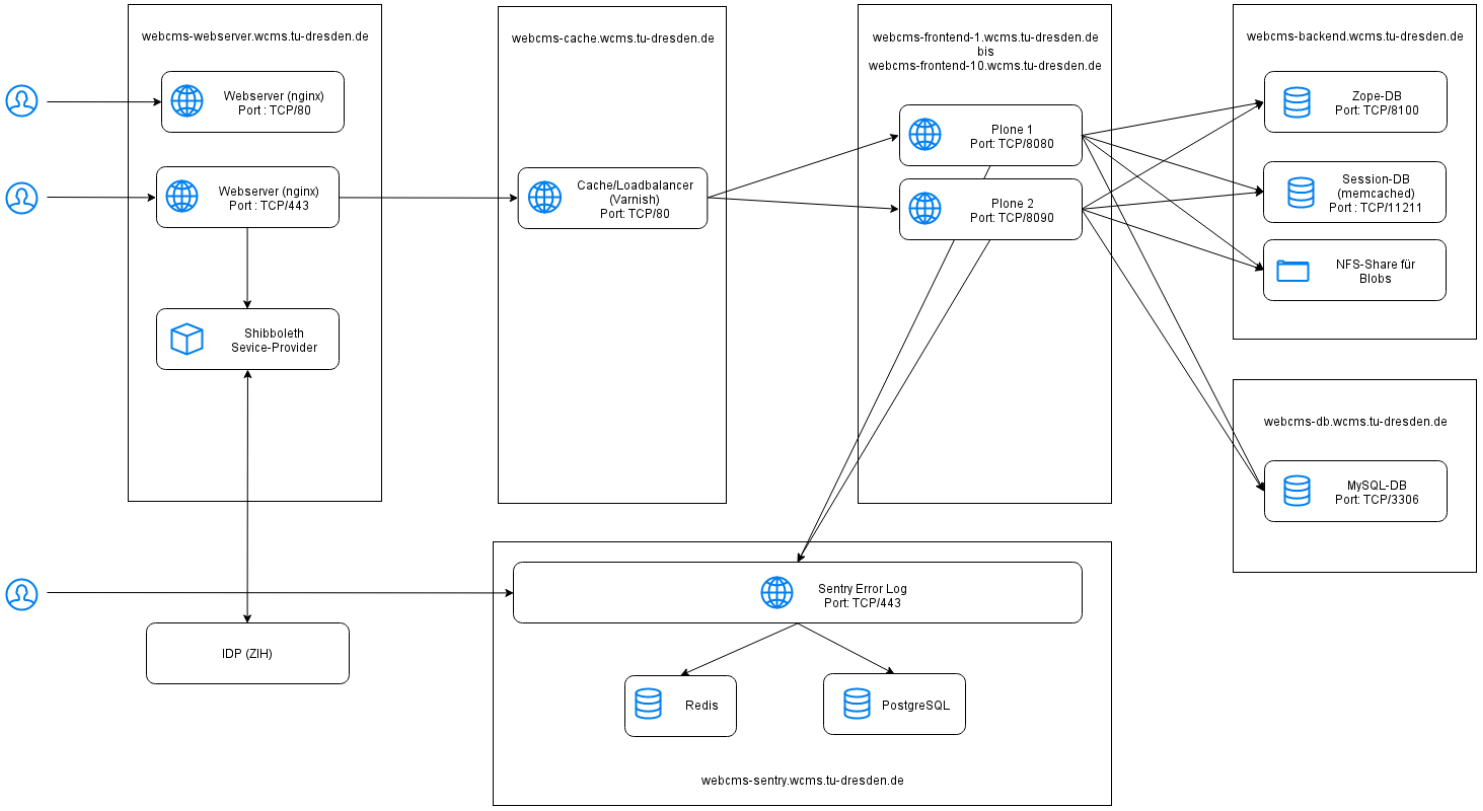
Multilingual Fields with raptus.multilanguagefields


TECHNISCHE UNIVERSITÄT DRESDEN
TU Dresden
Language
Search
Internal
MEDIEN ZENTRUM
THE UNIT
RESEARCH
SERVICES
FURTHER EDUCATION
Sebastian Gottfried [Manager]
Contents
View
Edit
Administrative functions
Add new
More Options
State: Published
Up one level
Media Centre
UPLOAD FILES
SORT FOLDER BY...
Select: All
Order
Title
Size
State
Actions
Medienzentrum (MZ)
1 KB
Apr 04, 2017 10:18 AM
The Unit
1 KB
Sep 26, 2016 11:10 AM

Key Figures

600k content objects
21 GB in ZODB filestorage
450 GB blobs
100 GB traffic per day
2.5 million page views per month
2500 active editors

Architecture



1. Challenge: Content Migration

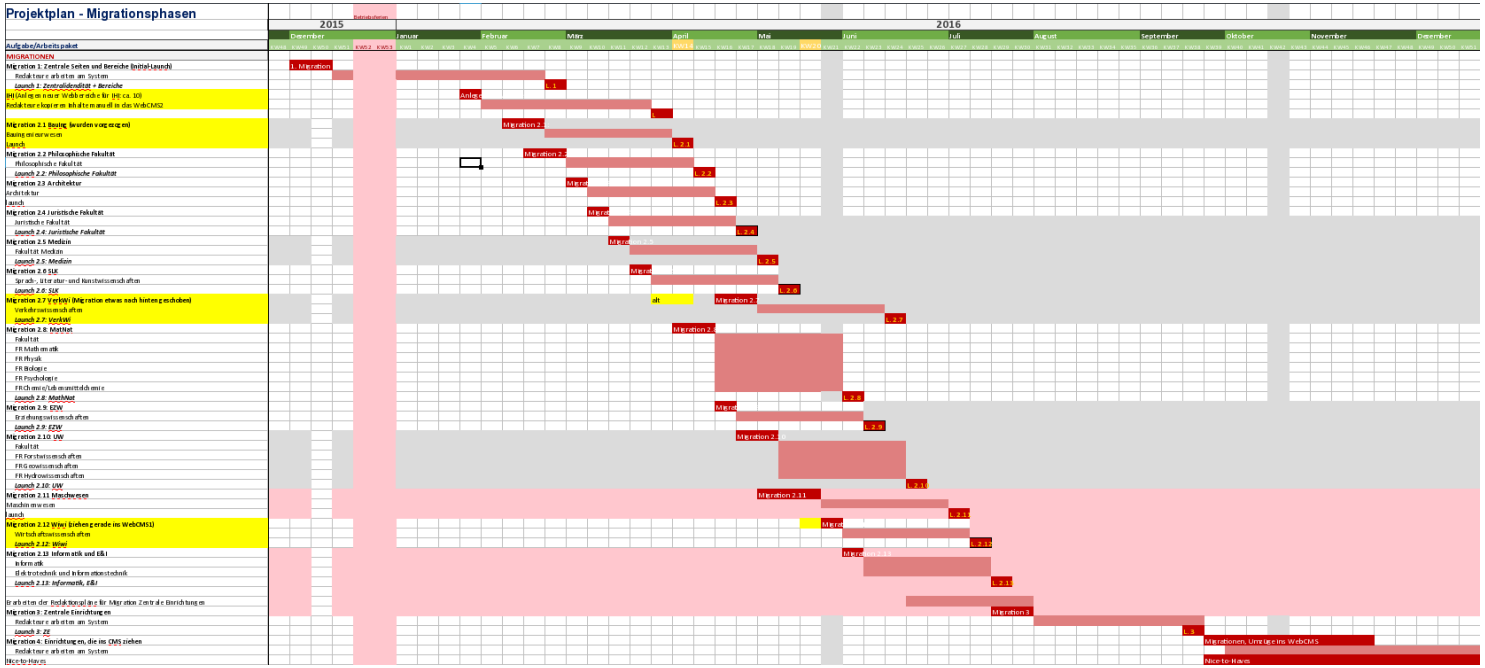
~ 500k million legacy content objects

Plone 2.1 without Archetypes

no `collective.transmogrifier` for us

The Migration Process

Executed in batches over the course of 1.5 years



The Migration Process

Simple custom-developed process with CLI scripts

1. Export to JSON + plain files for BLOB
2. Run transformations on JSON data
   - Map content types and field data to Plone's archetypes
   - Archive unwanted content
   - HTML clean up
3. Import in target system
   - At the designated place
   - Setup redirects
   - Special import folders for the remaining content
4. Editors finalize content
5. Section goes live

2. Challenge: User Management

Editorial team with more than 2500 members

Many of them are novice computer users

Limiting the actions an user can do is important

Body Text *

English

Standard

Image

Tempore labore debitis id ipsum. Vitae accusantium veritatis ullam sunt enim ut aliquam. Error explicabo accusamus explicabo autem qui aliquam assumenda. Omnis ad omnis nihil inventore facilis magni. Facere harum commodi quos. Cumque facilis ducimus voluptatum voluptas quia.

Est doloribus nulla dolores aut deleniti

Training

All editors had the opportunity to attend a half or full day training

83 training sessions

This was most our important communication channel

Direct link to user satisfaction

Login

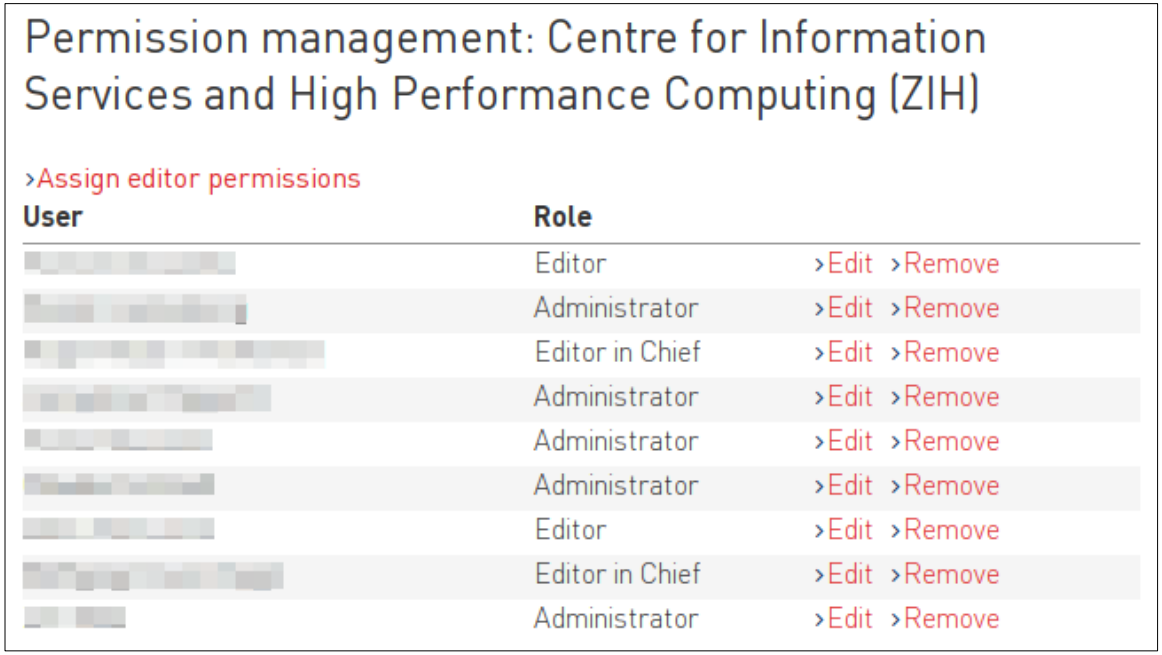
SSO setup with Shibboleth SP, NGINX and Products.AutoUserMakerPASPPlugin

Account data from central LDAP directory

Member role mapped to specific affiliation

Role Management

Local group management with `collective.workspaces`



Permission management: Centre for Information Services and High Performance Computing (ZIH)

›Assign editor permissions

User	Role		
[redacted]	Editor	›Edit	›Remove
[redacted]	Administrator	›Edit	›Remove
[redacted]	Editor in Chief	›Edit	›Remove
[redacted]	Administrator	›Edit	›Remove
[redacted]	Administrator	›Edit	›Remove
[redacted]	Administrator	›Edit	›Remove
[redacted]	Editor	›Edit	›Remove
[redacted]	Editor in Chief	›Edit	›Remove
[redacted]	Administrator	›Edit	›Remove

Role Management

We use the `native-groups-2` branch of `collective.workspaces`

No more catalog queries in permission checks

Strict Content Hierarchy

```
Root
    Section
        Faculty
            Institute
                Chair
                    News
                    Events
                    Page A
                    Folder
                        Page B
                    Resources
                        Images
                        Files
```

Enforced by addable content types and add permissions

Plone Caveat: Delete Permission

Expected behavior: permission controls which objects can be deleted

Actual behavior: permission controls in which container objects can be deleted

`collective.deletepermission` to rescue

3. Challenge: Consistent Performance

Our humble goal: keep load times below 1 sec

Extra complication: most users are authenticated

- Caching only of limited use
- Cache hit rate in Varnish ~ 25%

We just have to make the CMS go fast.

Common Pitfall: Rendering in TAL Conditions

Imagine you want to make an existence check for a content object.

Bad: `tal:condition="some_obj"`

Good: `tal:condition="nocall: some_obj"`

Also ok: `tal:condition="python: some_obj"`

ZEO Setup

Increased cache sizes for ZODB client caches

Persistent cache files

- Reduce time to rollout update by 50%

```
[zeoclient]
# ...
zeo-client-client = ${:_buildout_section_name_}
```

Session Store

Plone has two options:

- Sessions in temp storage: sessions limited to one Zope client
- Sessions in Filestorage: too slow for big sites

`collective.beaker` with memcached as backend

Just make sure to use `Beaker` ≥ 1.8.0

- Previous versions leaked connections to memcached

Be Careful with archetypes.schemaextender

We had to write a schema modifier for image field of the Image type

File and Image share the same schema object

Our „solution“: copy schema in schema modifier before changing fields

Problem: schema modifiers run on any field access and copying is expensive

Now we are back to regular content type classes

Dexterity behaviors look great

Useful Tools

haufe.longrunnningrequests

```
2017-07-25T06:27:10 WARNING RequestMonitor.DumpTrace Long running request
Request 108593 "/VirtualHostBase/https/tu-dresden.de/TUD/VirtualHostRoot/mz/die-
    einrichtung/team" running in thread 140460101805824 since 10.8283619881s
Python call stack (innermost first)
  Module plone.app.theming.transform, line 194, in transformIterable
  Module plone.transformchain.transformer, line 49, in __call__
  Module plone.transformchain.zpublisher, line 74, in applyTransform
  Module plone.transformchain.zpublisher, line 85, in applyTransformOnSuccess
  Module zope.interface.adapter, line 585, in subscribers
```

Piwik

PAGE URL	▼ PAGEVIEWS	UNIQUE PAGEVIEWS	BOUNCE RATE	AVG. TIME ON PAGE	EXIT RATE	AVG. GENERATION TIME
studium	20,440	15,562	55%	00:01:22	42%	1.16s
/index	6,881	4,528	35%	00:01:59	49%	0.57s
zih	5,708	4,213	40%	00:01:50	42%	1.18s
tu-dresden	5,224	3,740	49%	00:01:32	41%	1.18s
karriere	4,180	2,837	30%	00:01:05	23%	0.97s
usz	3,265	2,459	12%	00:01:21	40%	0.66s
zlsb	2,678	1,696	35%	00:02:00	36%	0.9s
/studium	1,435	927	18%	00:00:42	13%	1.07s
intern	1,215	842	20%	00:01:28	18%	0.78s

Portal Catalog Query Report

Contents | Catalog | Indexes | Metadata | Advanced | Query Report | Query Plan | Properties

Plone Catalog Tool at /portal_catalog

The **query report** shows catalog queries that perform slowly. For each index there's an additional entry for the time the intersection of the index result with the result by the other indexes took. These are marked with a *#intersection* postfix. The time reported for the index is the sum of the intersection time and the time the index itself took. Subtract the intersection time, if you want to know the pure index time.

Mean duration [ms]	Hits	Query key	Recent
5319.69	4	('SearchableText', 'allowedRolesAndUsers', 'effectiveRange', 'path', 'portal_type', 'sort_on', 'sort_order')	472.22ms [allowedRolesAndUsers#intersection: 0.01ms, portal_type#intersection: 0.01ms, effectiveRange#intersection: 0.01ms, SearchableText#intersection: 0.01ms, effectiveRange: 0.18ms, portal_type: 0.49ms, path#intersection: 1.27ms, allowedRolesAndUsers: 5.36ms, SearchableText: 47.85ms, sort_on: 50.00ms, path: 368.23ms,]
4410.53	1	('allowedRolesAndUsers', 'is_folderish', 'path', 'portal_type', 'sort_on')	4410.53ms [portal_type#intersection: 0.01ms, allowedRolesAndUsers#intersection: 0.02ms, sort_on: 0.31ms, path#intersection: 0.53ms, path: 0.59ms, portal_type: 1.79ms, is_folderish#intersection: 5.88ms, allowedRolesAndUsers: 58.47ms, is_folderish: 4349.20ms,]
1873.96	1	('allowedRolesAndUsers', 'path')	1873.96ms [allowedRolesAndUsers#intersection: 0.02ms, path#intersection: 9.21ms, allowedRolesAndUsers: 618.93ms, path: 1241.19ms,]
1515.83	1	('allowedRolesAndUsers', 'effectiveRange', 'end', 'object_provides', 'path', 'show_all', 'show_inactive', 'sort_on', 'start'	1515.83ms [path#intersection: 0.01ms, allowedRolesAndUsers#intersection: 0.02ms, start#intersection: 0.03ms, path: 0.06ms, sort_on: 0.06ms, object_provides#intersection: 0.32ms, object_provides: 0.37ms, end#intersection: 3.88ms, review_state#intersection: 3.94ms, review_state: 3.97ms, start: 5.02ms, effectiveRange#intersection: 7.40ms, effectiveRange:

collective.profiler

```
>>> stats.sort_stats('cumtime').print_stats('listActionInfos')
Thu Oct 19 11:12:31 2017    fc-table-v2.pstats

         1681681 function calls (1641420 primitive calls) in 6.604 seconds

   Ordered by: cumulative time
   List reduced from 682 to 2 due to restriction <'listActionInfos'>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
       16    0.000    0.000    1.512    0.095 ActionsTool.py:43(listActionInfos)
       15    0.000    0.000    0.000    0.000 TypesTool.py:52(listActionInfos)
```

4. Challenge: Quality Assurance

We deploy every two to four weeks to production

Manual testing is time consuming

We schedule two full days for this

Need for automation is obvious

Automated UI Testing

plone.app.robotframework

Compulsory for all new functions

CI system runs tests each night

S	W	Name ↓	Last Duration	Robot Results
		webcms2-tud.addons.admin-develop	20 min	16 / 24 passed
		webcms2-tud.addons.chat-develop	5 min 33 sec	8 / 8 passed
		webcms2-tud.addons.ckeditorplugins-develop	7 min 4 sec	16 / 16 passed
		webcms2-tud.addons.datagridfield-develop	3 min 59 sec	1 / 1 passed
		webcms2-tud.addons.protection-develop	5 min 10 sec	6 / 6 passed
		webcms2-tud.addons.redirect-develop	6 min 54 sec	8 / 8 passed
		webcms2-tud.boxes.base-develop	3 min 49 sec	1 / 1 passed
		webcms2-tud.boxes.webcms-develop	4 hr 56 min	219 / 222 passed
		webcms2-tud.content.webcms-develop	16 min	35 / 35 passed
		webcms2-tud.profiles.webcms-develop	18 min	14 / 14 passed
		webcms2-tud.theme.webcms2-develop	25 min	39 / 39 passed
		webcms2-wildcard.foldercontents-develop	6 min 19 sec	13 / 13 passed

plone.app.robotframework

Benefits

- Good for catching general problems
- Nice logs
- Screenshots

Issues

- Often not specific enough
- Runtime
- Selenium bugs

Thank you!

@plone_tudresden
@sebasgo86