static file cache

Static file caching using realurl, mod_rewrite and mod_expires.

. . . It slows down the warming of the earth.

Michiel Roos

Netcreators

netcreators

TYPO3

What does it do?

- Ehrm . . . it caches static files?
- I mean . . . it statically caches static pages?
- Eh . . . it generates static html files from static pages. If a static html file exists, mod_rewrite will redirect the visitor to the static page.
- This means that TYPO3 will not be loaded at all.
- Your server will have less work to do and will use less power. This helps to keep our earth cool ;-)

What's in it for me?

23000 %

speed improvement for static pages

Key features

- it's fscking fast
- transparent to the user
- works in the existing domain and port
- sends cache headers

Requirements

required

- apache
- mod_rewrite

recommended

- realurl
- mod_expires

in the beginning

- Tim Lochmüller: fl_staticfilecache (cheers Tim! ;-)
- the 'not so good':
  - xclasses
  - not my kind of code
  - backend module? But why?
  - need to actively clear cache
  - cache headers don't work with static html

The GOOD!

- Using mod_rewrite is a great idea!
- TYPO3 is very good at serving dynamic content
- Apache is very good at serving static html files
- Impressive speed improvement possible

Questions

- What can be cached?
- How can we invalidate / clear cache?
- How can we make cache headers work?
- What hooks exist?
- Where are they?
- When are they called?
- Does it work?
- Does it really work?

Shadow TYPO3

- create hook:
  - insertPageIncache
- clear hooks:
  - clearCachePostProc
  - clearPageCacheEval
  - tslib_fe-PostProc
- catch Ctrl+Shift+Reload:
  - tslib_fe-PostProc

How does it work?

- cache pages to static html file
- use mod_rewrite to:
  - check if a static html file exists, if it does . . .
  - redirect the request to the static html file, otherwise . . .
  - fall through to the next rewrite rule.

What does it look like?

- RewriteCond -> RewriteRule [L]
- RewriteCond %{DOCUMENT_ROOT}/typo3temp/tx_ncstaticfilecache/%{HTTP_HOST}/%{REQUEST_URI}index.html -f
- RewriteRule .* typo3temp/tx_ncstaticfilecache/%{HTTP_HOST}/%{REQUEST_URI} [L]

Other conditions

- RewriteCond %{HTTP:Pragma} !no-cache
- RewriteCond %{HTTP:Cache-Control} !no-cache
- RewriteCond %{HTTPS} off
- RewriteCond %{HTTP_COOKIE} !be_typo_user [NC]
- RewriteCond %{REQUEST_METHOD} GET
- RewriteCond %{QUERY_STRING} ^$

Enabling cache headers?

- Static files don't send cache headers.
- If you are using mod_rewrite, chances are . . .
- . . . you also have access to mod_expires.

Mod Expires

- If realurl is used we have a cache directory structure resembling the URI.
- Every cached file lives in it's own directory.
- Use the timeout value for the page . . .
- . . . to write a .htaccess file for every index.html.

```
<IfModule mod_expires.c>
  ExpiresActive on
  ExpiresByType text/html M60
</IfModule>
```

Cleaning up . . .

- Once the static file exists, it will stay.
- Eventually it will expire.
- When the browser requests a fresh page, it will still get the old one.
- Unless we manually remove the static file after it has expired.

The numbers

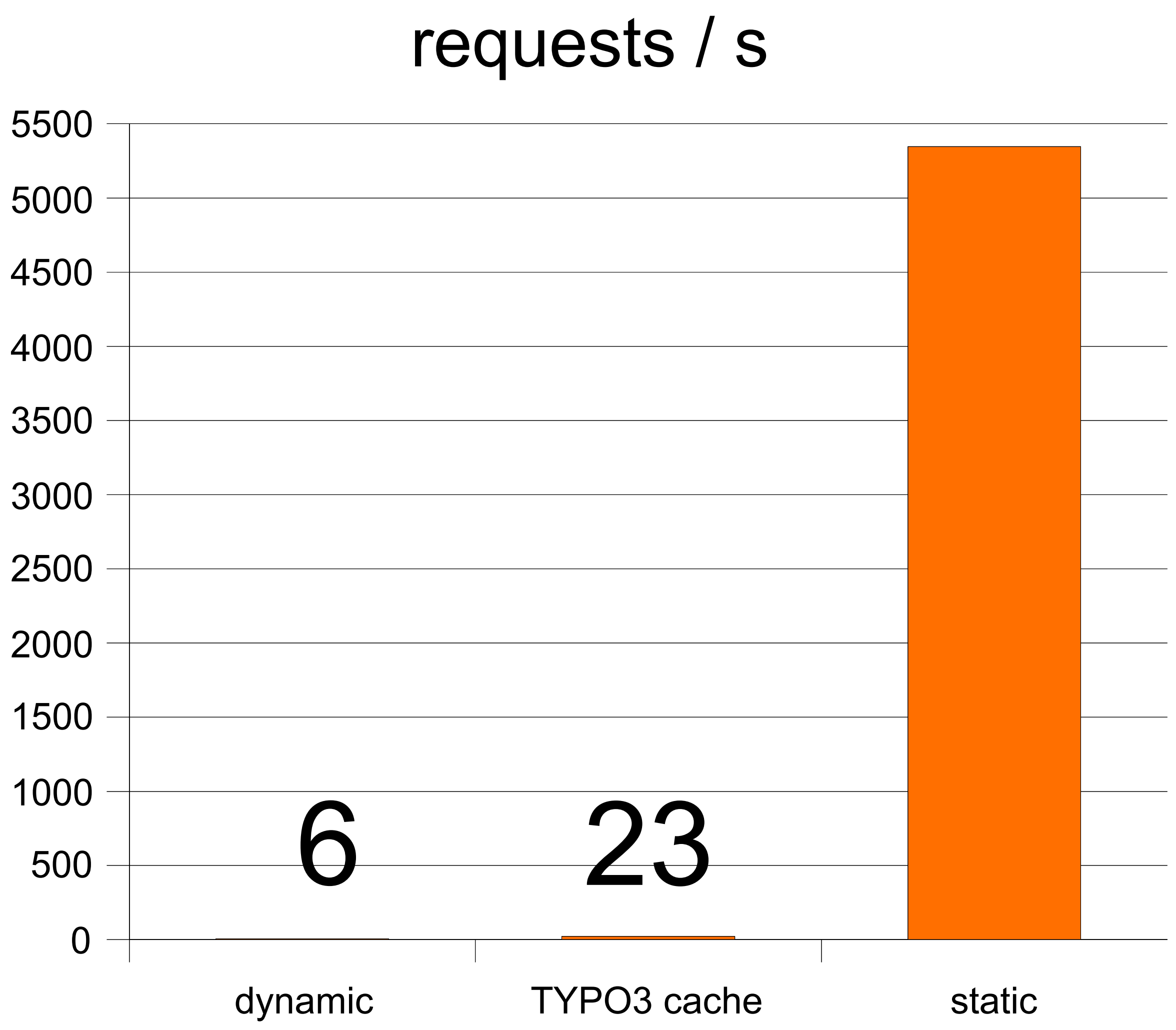
- ab
- apache bench
- try this on your moms server!

Server torture

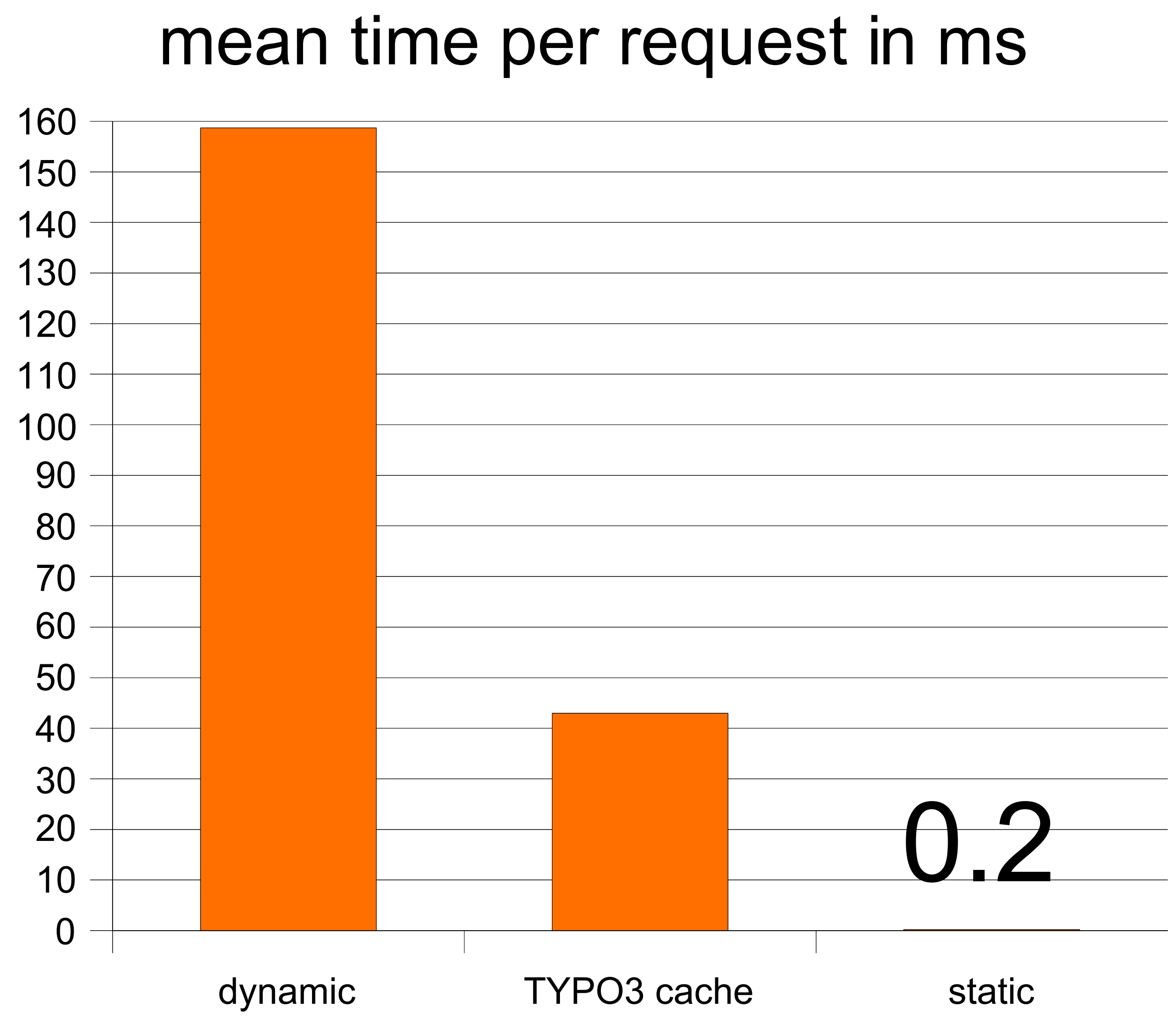
- ab -c 100 -n 1000 http://www.fictive.org/gpl-short/
- dynamic
- TYPO3 cache
- static

	requests / s	time / request ms	transfer rate Kbps	test time s
dynamic	6.3	158.68	83.82	158.68
TYPO3 cache	23.28	42.96	283.13	42.96
static	5346.62	0.19	71805.12	0.19

Serve more requests per second



Serve requests faster



Performance increase factor

230

times as quick

Let's see that again!

23000 %

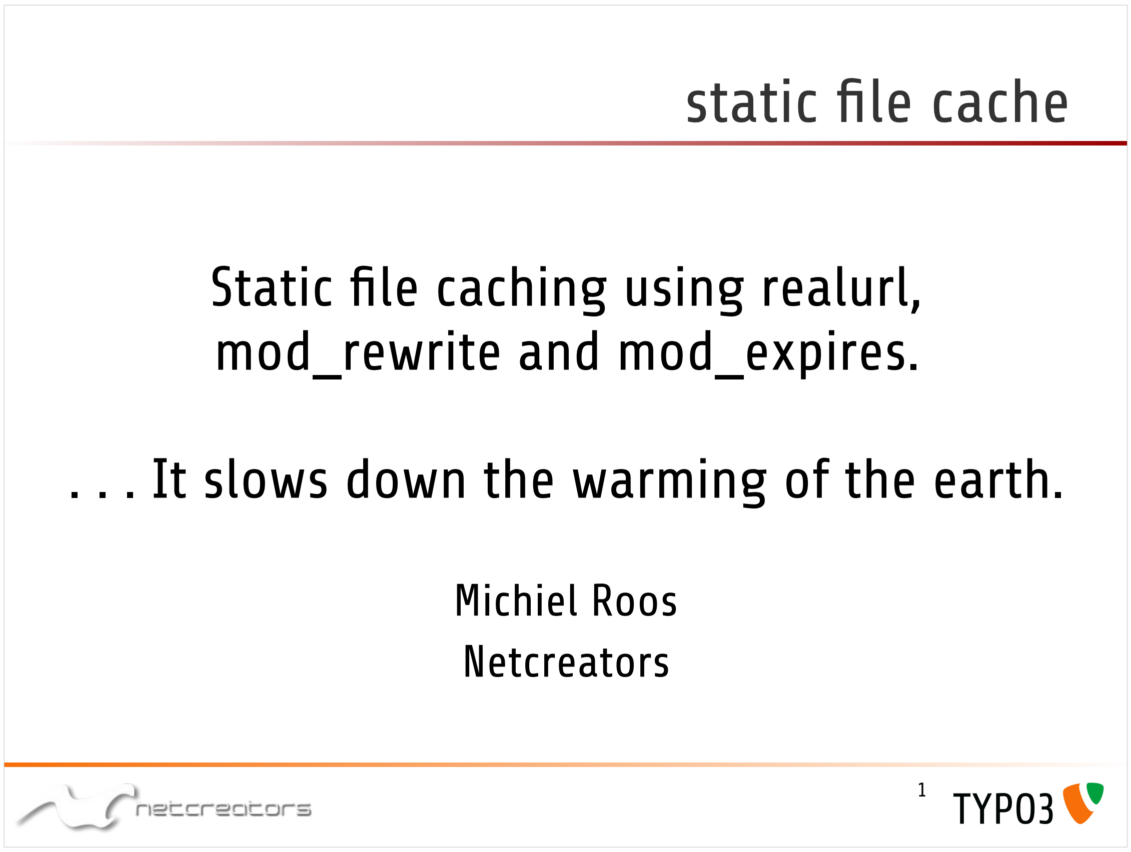
speed improvement

I have a dream . . .

- That one day . . .
- . . . all static file cache extensions . . .
- . . . will become one!
- called: 'staticfilecache'
- me scratch!

Tools

- eclipse
- svn
- firefox + live http headers
- cc_devlog
- grep, sed et al



static file cache

Static file caching using realurl, mod_rewrite and mod_expires.

. . . It slows down the warming of the earth.

Michiel Roos

Netcreators

netcreators 1 TYPO3

Recently Tim Lochmuller tranferred the key to netcreators.

Kaspar recently released staticpub

Benjamin Mack released bestfilecache

What does it do?

- Ehrm . . . it caches static files?
- I mean . . . it statically caches static pages?
- Eh . . . it generates static html files from static pages. If a static html file exists, mod_rewrite will redirect the visitor to the static page.
- This means that TYPO3 will not be loaded at all.
- Your server will have less work to do and will use less power. This helps to keep our earth cool ;-)

What's in it for me?

23000 %

speed improvement for static pages

Key features

- it's fscking fast
- transparent to the user
- works in the existing domain and port
- sends cache headers

Requirements

required

- apache
- mod_rewrite

recommended

- realurl
- mod_expires

in the beginning

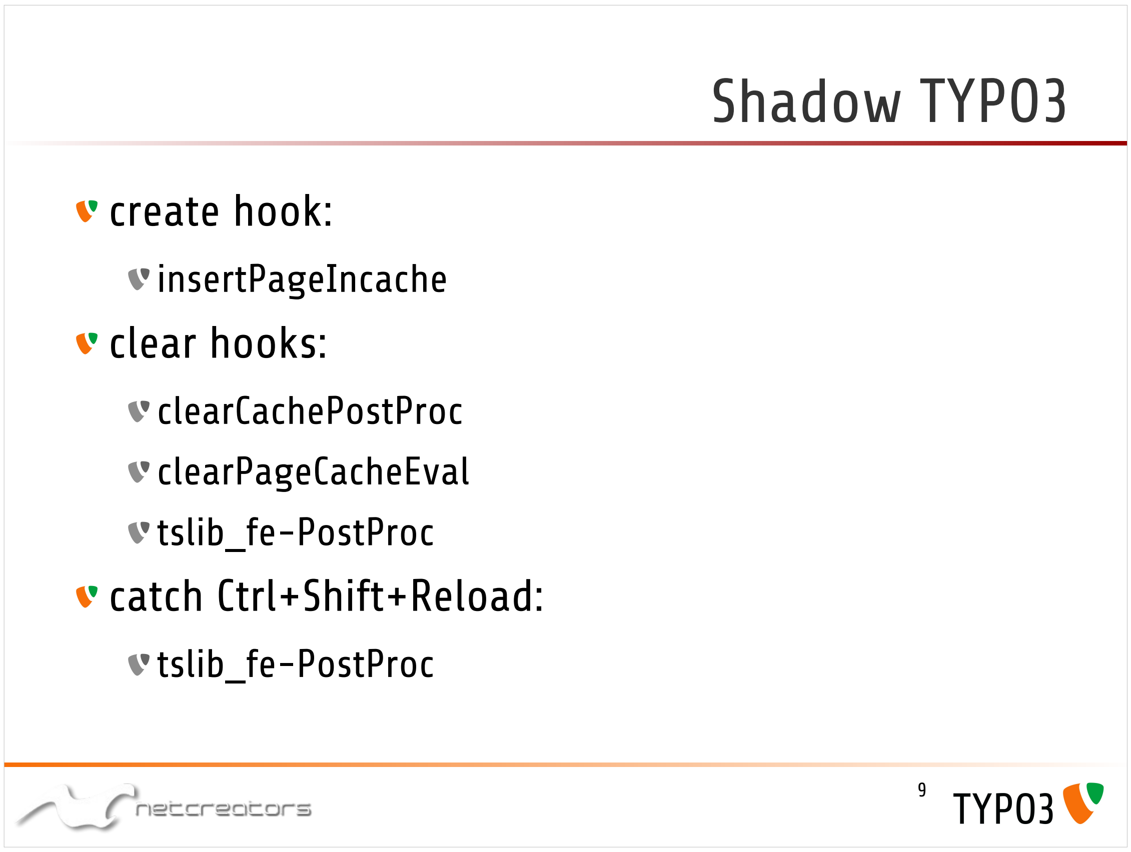
- Tim Lochmüller: fl_staticfilecache (cheers Tim! ;-)
- the 'not so good':
  - xclasses
  - not my kind of code
  - backend module? But why?
  - need to actively clear cache
  - cache headers don't work with static html

The GOOD!

- Using mod_rewrite is a great idea!
- TYPO3 is very good at serving dynamic content
- Apache is very good at serving static html files
- Impressive speed improvement possible

Questions

- What can be cached?
- How can we invalidate / clear cache?
- How can we make cache headers work?
- What hooks exist?
- Where are they?
- When are they called?
- Does it work?
- Does it really work?


Shadow TYPO3

- create hook:
  - insertPageIncache
- clear hooks:
  - clearCachePostProc
  - clearPageCacheEval
  - tslib_fe-PostProc
- catch Ctrl+Shift+Reload:
  - tslib_fe-PostProc

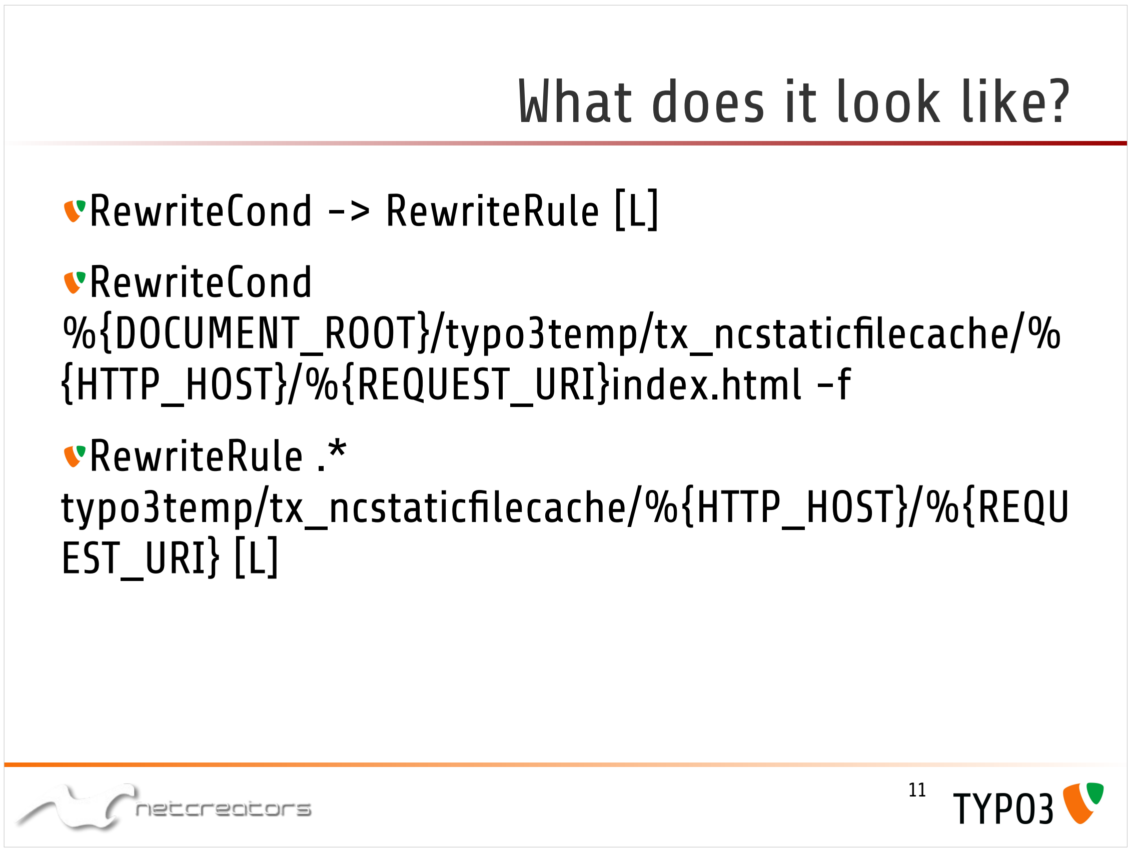
netcreators 9 TYPO3

What can we cache? When do we clear cache? We can follow TYPO3's behaviour.

If TYPO3 caches a page, we will create a static version, if TYPO3 clears the page cache, we will delete the static version.

How does it work?

- cache pages to static html file
- use mod_rewrite to:
  - check if a static html file exists, if it does . . .
  - redirect the request to the static html file, otherwise . . .
  - fall through to the next rewrite rule.



What does it look like?

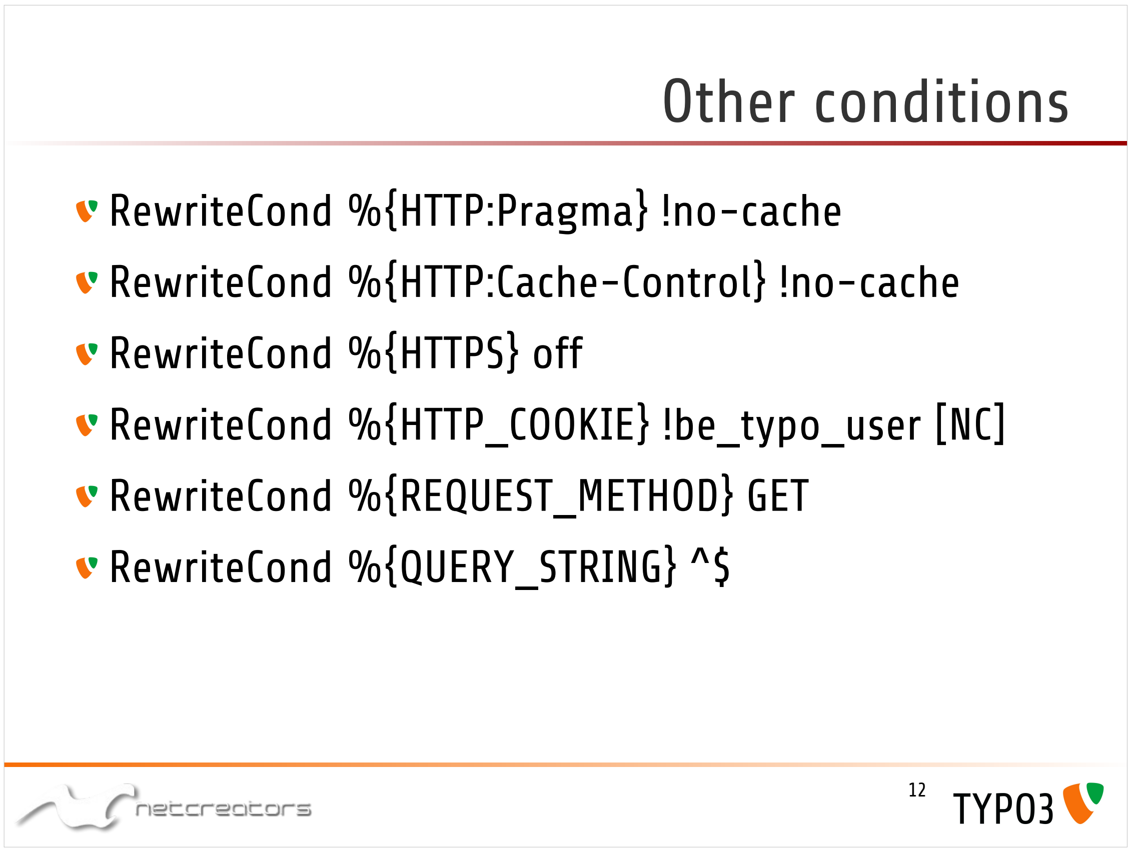
- RewriteCond -> RewriteRule [L]
- RewriteCond %{DOCUMENT_ROOT}/typo3temp/tx_ncstaticfilecache/%{HTTP_HOST}/%{REQUEST_URI}index.html -f
- RewriteRule .* typo3temp/tx_ncstaticfilecache/%{HTTP_HOST}/%{REQUEST_URI} [L]

netcreators 11 TYPO3

Mod rewrite works with rulesets consising of zero or more rewrite conditions followed by a rewrite rule.
If all conditions are met, the rule is executed.
If one of the conditions is not met, the ruleset is terminated. mod rewrite moves on to the next ruleset.
The Rewrite condition checks (amongst other things) the staticfilecache path to see if a static html file exists.
Other things it checks for:

```
RewriteCond %{HTTP:Pragma} !no-cache
RewriteCond %{HTTP:Cache-Control} !no-cache
RewriteCond %{HTTPS} off
RewriteCond %{HTTP_COOKIE} !be_typo_user [NC]
RewriteCond %{REQUEST_METHOD} GET
RewriteCond %{QUERY_STRING} ^$
```

If it does, it rewrites the request to the static file.



Other conditions

- RewriteCond %{HTTP:Pragma} !no-cache
- RewriteCond %{HTTP:Cache-Control} !no-cache
- RewriteCond %{HTTPS} off
- RewriteCond %{HTTP_COOKIE} !be_typo_user [NC]
- RewriteCond %{REQUEST_METHOD} GET
- RewriteCond %{QUERY_STRING} ^$

netcreators

TYPO3

Check for Ctrl+Shift+Reload

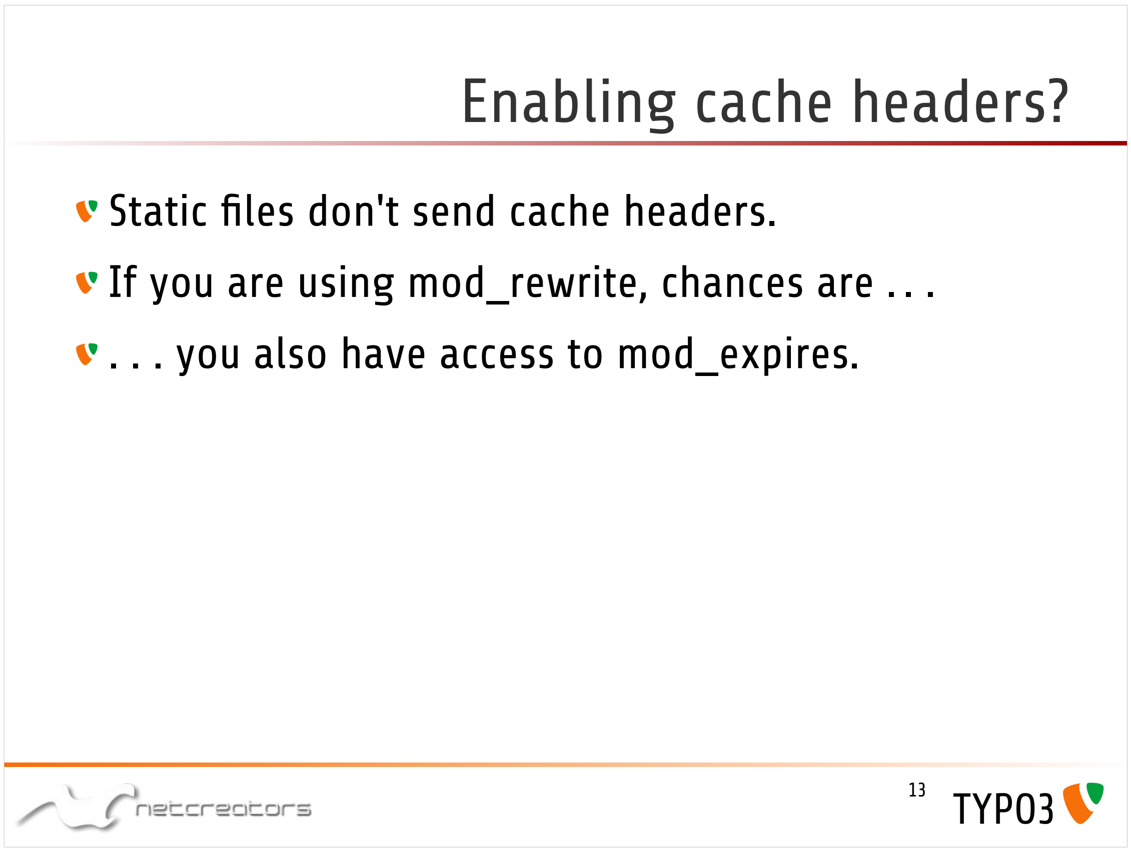
Don't cache HTTPS traffic by default. You may choose to comment out this option if your site runs fully on https. If your site runs mixed, you will not want https traffic to be cached in the same typo3temp folder where it can be requested over http.

NO backend user is logged in. Please not that the be_typo_user expires at the end of the browser session. So, although you have already logged out of the backend, you will still have to either restart your browser or remove the cookie manually for this rule to work.

We only redirect GET requests

We only redirect URI's without query strings

NC: non case sensitive



Enabling cache headers?

- Static files don't send cache headers.
- If you are using mod_rewrite, chances are . . .
- . . . you also have access to mod_expires.

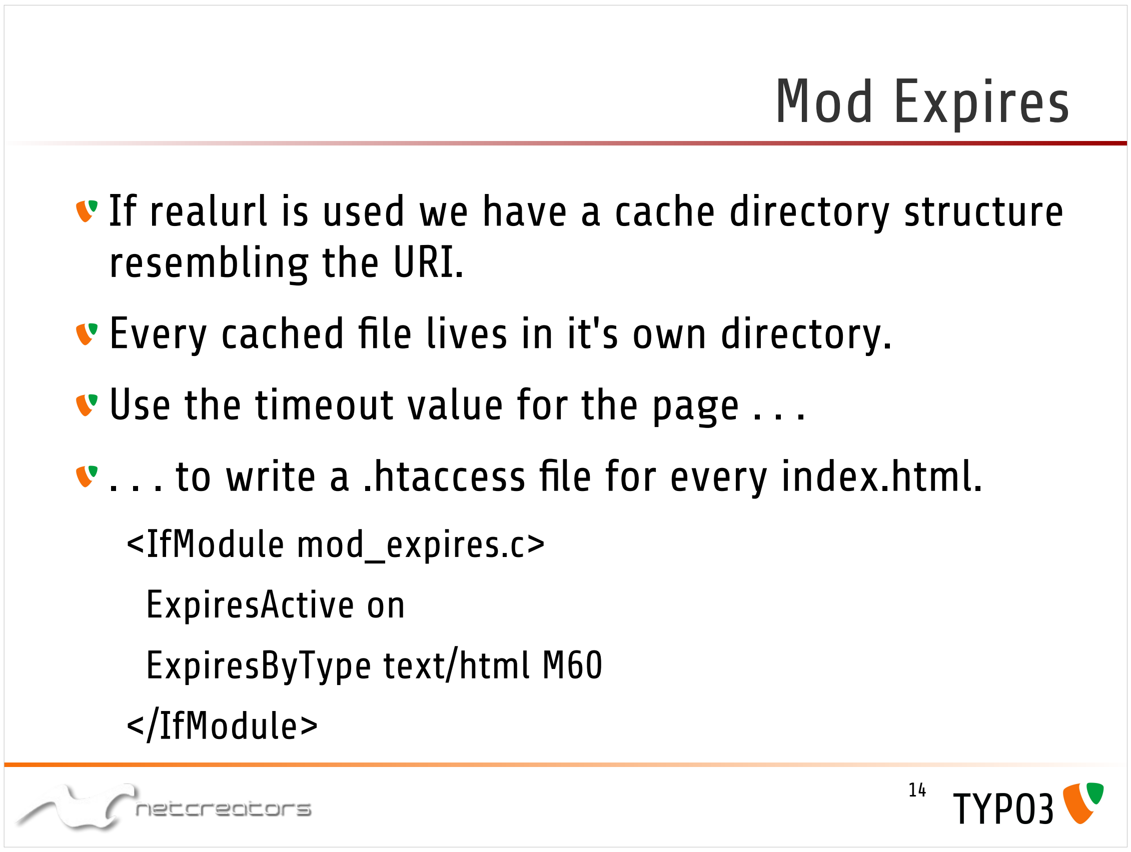
netcreators

TYPO3

Static files do not send cache headers. Of course there is meta: expires. But it's a hassle. May be implemented at a later stage if tests show it works as it should.

Use mod_expires to send the cache headers.

This is possible because of realurl.



Mod Expires

- If realurl is used we have a cache directory structure resembling the URI.
- Every cached file lives in it's own directory.
- Use the timeout value for the page . . .
- . . . to write a .htaccess file for every index.html.

```
<IfModule mod_expires.c>
  ExpiresActive on
  ExpiresByType text/html M60
</IfModule>
```

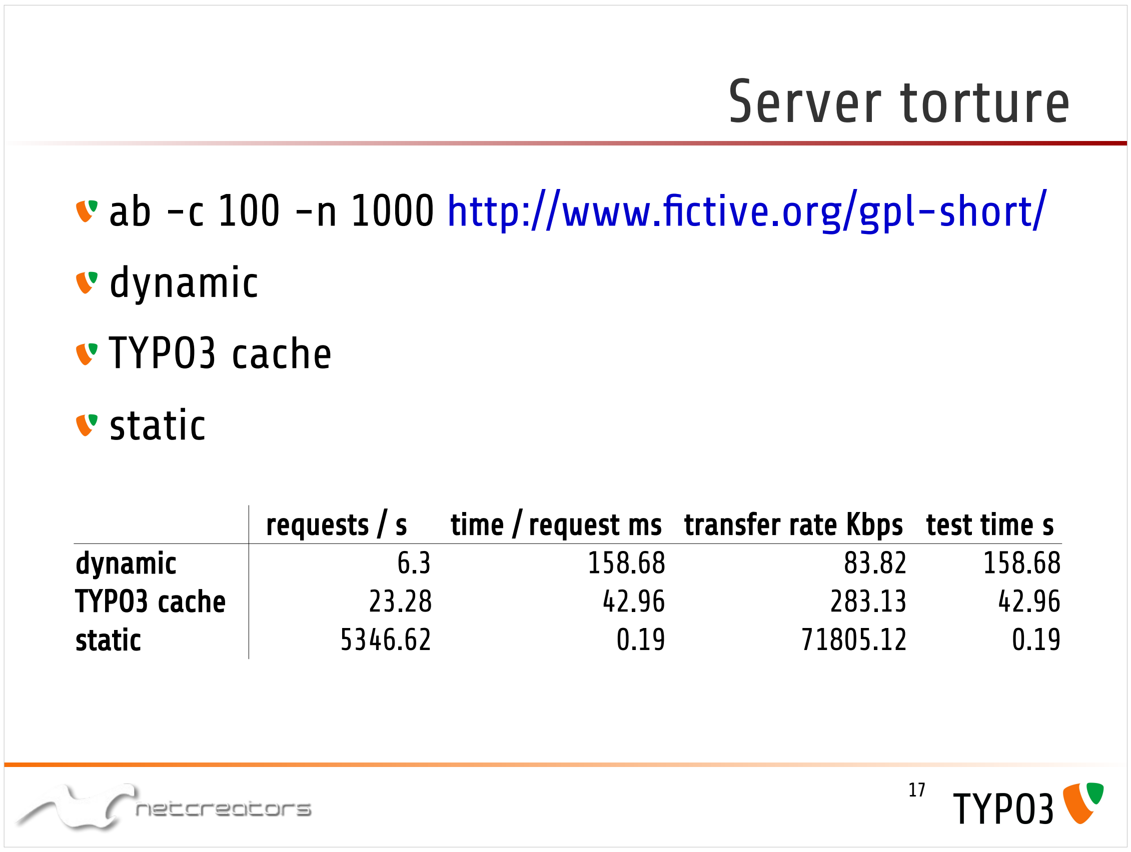
We use the page timeout value to write a htaccess file.
We use the format 60M for example. This means that cache headers will be sent containing an expiry time of 'file modification time' + 60 seconds.
So as time progresses the cache headers sent for the file will send shorter expiry times.
In combination with the cleaner script we can build an up to date cache.
We select the expired pages by querying the pages table with: 'expires <='.$GLOBALS['EXEC_TIME']);
You need to run your cache cleaner script in a cron job with your lowest cache expiry time.

Cleaning up . . .

- Once the static file exists, it will stay.
- Eventually it will expire.
- When the browser requests a fresh page, it will still get the old one.
- Unless we manually remove the static file after it has expired.

The numbers

- ab
- apache bench
- try this on your moms server!



Server torture

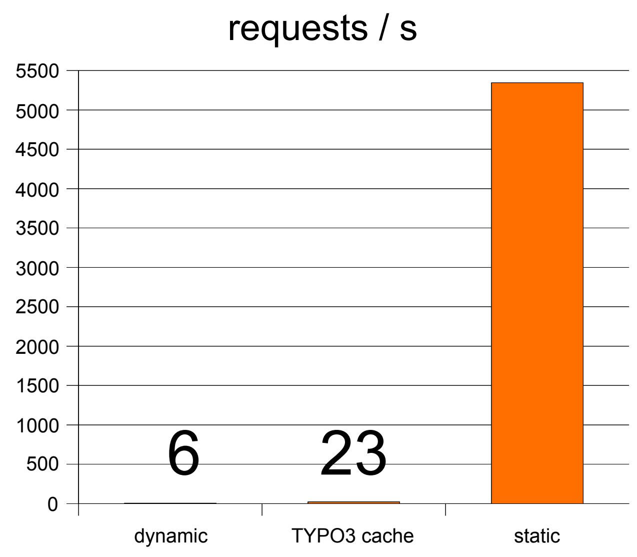
- ab -c 100 -n 1000 http://www.fictive.org/gpl-short/
- dynamic
- TYPO3 cache
- static

	requests / s	time / request ms	transfer rate Kbps	test time s
dynamic	6.3	158.68	83.82	158.68
TYPO3 cache	23.28	42.96	283.13	42.96
static	5346.62	0.19	71805.12	0.19

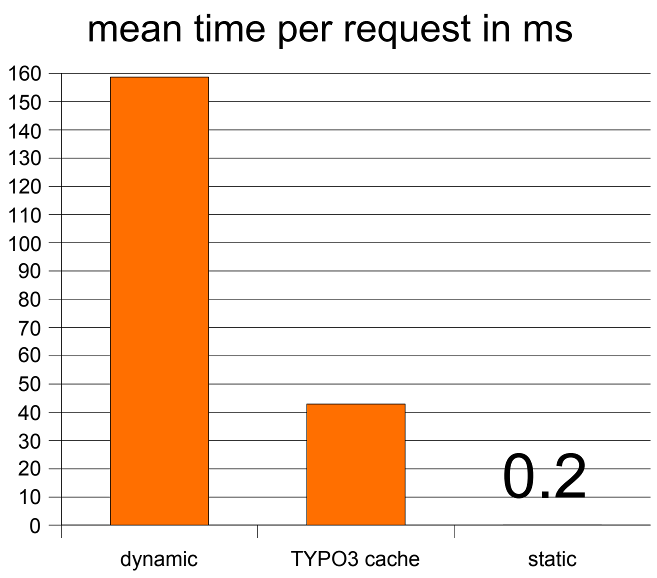
ab is a tool for benchmarking your Apache Hypertext Transfer Protocol (HTTP) server. It is designed to give you an impression of how your current Apache installation performs. This especially shows you how many requests per second your Apache installation is capable of serving.

-c concurrency
Number of multiple requests to perform at a time. Default is one request at a time.

Serve more requests per second



Serve requests faster



Performance increase factor

230

times as quick

Let's see that again!

23000 %

speed improvement

I have a dream . . .

- That one day . . .
- . . . all static file cache extensions . . .
- . . . will become one!
- called: 'staticfilecache'
- me scratch!

Tools

- eclipse
- svn
- firefox + live http headers
- cc_devlog
- grep, sed et al