Using XSL and mod_transform in Apache Applications

Paul Querna
chip@OutOfOrder.cc
What is XSL?

- Extensible Stylesheet Language (XSL)
- A family of Standards for XML by the W3C:
  - XSL Transformations (XSLT)
  - XML Path Language (Xpath)
  - XSL Formatting Objects (XSL-FO)
<?xml version="1.0"?>
<xsl:stylesheet
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version="1.0">

    <xsl:template match="/">
        <html>
            <head><title>A Message</title></head>
            <body>
                <h1>
                    <xsl:value-of select="message" />
                </h1>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
<?xml version="1.0"?>
<message>Hello World</message>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>A Message</title>
</head>
<body>
<h1>Hello World</h1>
</body>
</html>
Why is XSLT good?

- Mixing Data and Presentation is **bad**!
  - Keeps Data in a clean XML schema
  - Keeps the Presentation of this Data separate
- XSLT is XML
- Easy to Extend
- Put HTML or other Markups directly in the XSLT.
  - Easy for Web Developers to create a template
Why is XSLT bad?

- XSLT is XML
- Complicated XSLT can be slow
- Yet another language to learn
Where does Apache fit in this?

- Apache 2.0 has Filters!

Client → Input Filters → Handlers (Perl, PHP, Proxy, File) → Output Filters

- mod_include (SSI)
- mod_transform (XSLT)
- mod_deflate (gzip)
mod_transform

- Uses libXML2 and libXSLT from Gnome
  - C API
    - Doesn't depend on other Gnome Libs.
  - Provides:
    - EXSLT
    - XInclude
    - XPath
    - Xpointer
    - ... and more
Static XML Files

- AddOutputFilter XSLT .xml
- TransformSet /xsl/foo.xsl
  - Only if your XML does not specify a XSL File
- TransformOptions +ApacheFS
  - Uses Sub-Requests to find files
  - Makes mod_transform work like Apache AxKit
Dynamic Sources

• XML Content Types:
  – AddOutputFilterByType XSLT application/xml

• Controlled Content Types:
  – AddOutputFilterByType XSLT applicain/needs-xslt

• Works for Proxied Content, PHP, mod_perl, mod_python, CGI, SSI, etc.:
  – mod_filter
    • In 2.1 CVS!
PHP

• Content Type

• With a Patch
  - PHP can manually add output filters
    • apache_add_output_filter("XSLT");
  - List Active Filters
    • apache_get_output_filters();
<?php
apache_add_output_filter("XSLT");

echo "<?xml version='1.0'?>\n";
echo "<?xml-stylesheet type='text/xsl'
href='./demo.xsl'?>\n";
echo "<demo>\n";

filters = apache_get_output_filters();

foreach($filters as $filter) {
    echo "<filter>$t_val</filter>\n";
}

echo '</demo>';
?>
Apache Module

• Direct:
  - `ap_add_output_filter("XSLT", NULL, r, r->connection);`

• By Content Type:
  - `ap_set_content_type(r,"application/xml");`

• Passing a libXML2 Document Tree:
  - `mod_transform_XSLTDoc(r, doc);`
mod_svn_view

- Uses mod_transform
- Provides a web based view using the Native Subversion C libraries
### SVN View

[Repo List] - [gentoo]/apache-overlay

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>[no properties set]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Filename</th>
<th>Revision</th>
<th>Username</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>[dir] profiles</td>
<td>321</td>
<td>Hollow</td>
<td>* changed apache-mod...</td>
</tr>
<tr>
<td>[dir] scripts</td>
<td>336</td>
<td>beu</td>
<td>* new tool: - simple...</td>
</tr>
<tr>
<td>[dir] docs</td>
<td>269</td>
<td>stuart</td>
<td>Added web-handbook a...</td>
</tr>
<tr>
<td>[dir] dev-util</td>
<td>366</td>
<td>trapni</td>
<td>renamed subversion 1...</td>
</tr>
<tr>
<td>[dir] dist</td>
<td>369</td>
<td>vericgar</td>
<td>Fixing default confi...</td>
</tr>
<tr>
<td>[dir] www-apache</td>
<td>367</td>
<td>trapni</td>
<td>version bump</td>
</tr>
<tr>
<td>[dir] dev-libs</td>
<td>356</td>
<td>chip</td>
<td>APR-1.0 ebuilds now ...</td>
</tr>
<tr>
<td>[dir] net-www</td>
<td>368</td>
<td>Hollow</td>
<td>version bump</td>
</tr>
<tr>
<td>[dir] eclass</td>
<td>358</td>
<td>urilith</td>
<td>Cleaning up the ecla...</td>
</tr>
<tr>
<td>[dir] irc</td>
<td>347</td>
<td>Hollow</td>
<td>* assigned more spec...</td>
</tr>
<tr>
<td>[dir] dev-perl</td>
<td>258</td>
<td>trapni</td>
<td>did an s,extramodule...</td>
</tr>
<tr>
<td>[file] ChangeLog</td>
<td>159</td>
<td>mtindal</td>
<td>Final commit for tre...</td>
</tr>
<tr>
<td>[file] skel.apache1.ebuild</td>
<td>337</td>
<td>Hollow</td>
<td>* many many mod_* eb...</td>
</tr>
</tbody>
</table>
Performance Issues

- Parsing the XSLT with libXSLT is slow
  - mod_transform can cache the XSL Files
- Parsing the XML to a libxml2 is slow
  - Applications can pass a docPtr in memory
- Doing the Transform itself is slow
  - mod_cache can capture the output
Benchmarks!

- Your Millage May Vary
  - Test to your environment/hardware/software
No Caching

![Graph showing the performance of No Caching with respect to Clients and Requests/Second.](image-url)
XSL Static Cache Issues

- Does not see changed XSL Files
  - Not a big issue on production servers
  - Only sees changes on an Apache Restart
- 'Lowest Hanging Fruit' to improve performance
mod_cache

The diagram above illustrates the performance of different caching methods with varying numbers of clients. The x-axis represents the number of clients, while the y-axis shows the number of requests per second. The lines represent:

- **Blue** - No-caching
- **Red** - XSL Static Cache
- **Green** - mod_disk_cache

The graph shows that as the number of clients increases, the number of requests per second also increases, with the caching methods having a significant impact on performance. The green line, representing **mod_disk_cache**, demonstrates the highest performance, followed by the red line for **XSL Static Cache**, and the blue line for **No-caching**.
mod_cache Issues

• mod_cache:
  - Doesn't have any dependency checks
    • Changed content is not seen!
    • Changed XML Files are not noticed!
    • Changed XSL Files are not noticed!
  - Can be fixed
    • Etag Generation
      - In the ~/TODO
      - Slower, but should still be faster than running the XSL
Alternatives

- mod_php / mod_perl / mod_python
  - Can use libXSLT
- mod_ext_filter
  - Use an external Process
    - Performance?
- Apache AxKit
  - Apache 1.3 Only
  - Requires mod_perl
  - Cannot transform content from other Handlers
Alternatives

- mod_xmlns / mod_publisher
  - Still in development
  - Framework for a custom template system
- Modxslt2
  - Provides Extra Features
    - Access to Query Arguments and POST
  - ~10,000 LOC
Future Development

- Dependency Trees – mod_depends
  - Etags
  - LastModified
- Access to Query Arguments and POST
- Dynamic XSL Caching
Resources

- Slides, mod_transform, PHP Patch:
  - http://www.outoforder.cc
- LibXML2:
  - http://xmlsoft.org/
- LibXSLT:
  - http://xmlsoft.org/XSLT/
- Apache AxKit
  - http://axkit.org/
Resources

- mod_publisher / mod_xmlns
  - http://apache.webthing.com
  - http://apache.webthing.com/mod_publisher/

- Modxslt2
  - http://www.mod-xslt2.com/
Thanks!

- mod_transform co-authors:
  - Nick Kew
  - Edward Rudd

Questions?