Patches for libxslt/libxml

Carlo Contavalli

This document describes a couple patches to apply to libxml and libxslt that could be useful to improve mod-xslt2 overall performance and reliability, even if their absence does not compromise mod-xslt2 usability.

1. License and Copyright

This document is Copyright (C), Carlo Contavalli 2003, 2004.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of this license can be found at http://www.commedia.it/ccontavalli/license/FDL/en/index.html.

2. libxml - libxml.setglobalstate.diff

The setglobalstate patch allows an application to make use of the internal "xmlGetGlobalState" function provided by libxml2 and adds and adds a "xmlSetGlobalState".

libxml makes use of many global variables kept in a task specific area (when multithreading is enabled) to avoid race conditions with other threads. However, this is not enough for mod-xslt: many apache modules share the same thread of execution and thus share the same "task specific" libxml data. To avoid interactions with other modules using libxml, mod-xslt saves the global execution state and restore it for each processed request. The process of "storing" and "restoring" is in facts a "copy" of a memory area (using memcpy). However, libxml internally uses a much more efficient mechanism that would allow mod-xslt to "store" and "restore" a single pointer on most POSIX systems (using POSIX threads).

The highlited patch allows an application (like mod-xslt) to make use of those functions improving overall performance.

The patch is automatically detected by mod-xslt "configure" and should not cause any harm to other applications.

3. libxslt - libxslt.genericerror.diff

Internally, libxslt calls "xsltGenericError" to signal an error to the application making use of the library.

However, sometimes, "xsltGenericError" gets called by libxslt with the arguments of the function "xsltGenericDebug" instead of its own, causing a SEGFAULT in mod-xslt (which doesn't expect the wrong arguments to be passed - it would cause a SEGFAULT on most applications).

Those "xsltGenericError" calls should never be reached in the normal path of execution of the library and of an application. However, it came out that they are sometimes reached, usually due to bugs in libxml2 or in the xslt handlers set by mod-xslt (quite unlikely :).

A sort of harmless "parachute" has been employed in mod-xslt to detect this situation *most* of the times. However, this mechanism doesn't guarantee the error will be avoided *all* the times.

To completely avoid this problem, you should patch the library with the provided .diff (or use the configure parameter described below).

The patch was created by running something like:

```
find . -type f -name '*.c' |xargs perl -pi -e \
's/xsltGenericError\(xsltGenericDebugContext/xsltGenericError\(xsltGenericErrorContext/'
```

from the libxslt source tree.

3.1. configure - --enable-libxslt-hack

Another solution to patching the library is to use "--enable-libxslt-hack". Enabling this option to configure, mod-xslt will be configured to setup a fake "debug" error function, which uses the same parameters as the standard error functions.

That way, the problem is completely avoided without needing any further action.

Note, however, that by enabling this option you enable libxslt debugging facilities, which may potentially slow down the parsing.

4. libxslt - libxslt.fallback.diff

During debugging, it came out that most of those errors that used the wrong "arguments" were triggered by placing a "<fallback>" tag in the wrong position.

When such a tag is found, "libxslt" will print an error message due to the fact it reaches an unexpected position in the code.

However, if you use extension tags provided by mod-xslt and the fallback handler, you will see quite a bounce of these errors. If the library was not patched with libxslt.genericerror.diff or mod-xslt not compiled with

"--enable-libxslt-hack", then each of these error would cause a SEGFAULT (however, this is not my fault), otherwise, an annoying warning is outputted in the logs.

To avoid this warning filling the logs, you can apply libxslt.fallback.diff, which, conforming with "REC-xslt-19991116.html#fallback" will make libxslt ignore the presence of such a tag.

4.1. configure - --enable-fallback-wraparound

Another solution to patching the library is to compile mod-xslt passing "--enable-fallback-wraparound" as an argument to configure.

Enabling this option will cause some more code to be compiled in mod-xslt. This code will take care of "stripping down" the <fallback tags from "legal" positions made illegal by mod-xslt usage of the library.

Eg, the patch solves the problem once and forever. This option solves the problem when it is triggered by mod-xslt, which means every time a <fallback tag is found inside a mod-xslt extension tag.