Subject: Re: xsl transformation slow
Posted by braingeyser on Mon, 19 Mar 2007 02:18:19 GMT
View Forum Message <> Reply to Message

Quote:(a) Avoid such large record sets.

Never underestimate the ability of an end user to do what should not be done.

Quote:
(b) Upgrade to PHP 5
(c) Upgrade your hardware.

This is not always an option for remotely hosted sites.

Actually, there is a very simple optimisation that will speed up the xsl transformation by a factor of around 300%.

I had a quick look at the std_std.data_field.xsl file and figured that the lions share of the processing would be done by the template called "display_horizontal" (for the list1 transaction pattern at least).

The thing that struck me right away was that inside the for_each loop a variable called "field" is declared and is selected starting from the root of the document every time. I've learned from hair-pulling experience that this is to be avoided whenever possible. It basically means that in order to create this variable, the ENTIRE xml document must be searched (which is what the "//" means) every time the variable is created. For large record sets this can be prohibitive. The obvious solution is to whittle down what needs to be searched before the variable is declared.

the offending line is this....

<xsl:variable name="field" select="//*[name()=$table][position()=$position]/*[name()=$fieldname]" />

Now most of that select changes on each iteration of the for_each loop due to the position() function, but the very beginning can be replaced by a variable declared outside the loop, e.g

<xsl:variable name="table_tmp" select="//*[name()=$table]" />

This replaces the corresponding code in the variable declaration for "field" like so...

<xsl:variable name="field" select="$table_tmp[position()=$position]/*[name()=$fieldname]" />

..and voila. A few very quick tests showed improvements from around 250-300% in xsl transformation times. I'm sure the same principle can be applied in other places for an even greater increase. The important thing to remember is to avoid using "//" repetitively if at all possible.

My apologies if this sounded like a lecture for those who know how xsl works. It was written for those who don't. Obviously Tony will know what I'm talking about which is all that counts

For those interested in learning more about XSL (and you should, it's very cool) I can thoroughly recommend a book I bought on the subject called "Beginning XSLT 2.0: From Novice to Professional" by Jeni Tennison. A most excellent tutorial style book that is worth its weight in gold.

Cheers,
Craig
