Subject: Re: xsl transformation slow
Posted by braingeyser on Mon, 19 Mar 2007 07:25:11 GMT
View Forum Message <> Reply to Message

OK, I've found one further significant increase in speed, but it's entirely dependant on the answer to this question...

Is the relative position of the <structure> tag in any xml document that uses the "display_horizontal" template in it's xsl transformation ever going to be anything other than /root/structure ? If so then never mind.

If not then read on...

Again, the use of "//" has crippled the performance of the transformation. The xsl:for-each loop I referred to in my original post has the following select criteria...

<xsl:for-each select="//structure/*[name()=$zone]/row/cell[@field]">

what this selection does is to search the entire xml document,including the table data, for a <structure> tag at any level however deeply buried, before applying the remaining selection criteria of...

/*[name()=$zone]/row/cell[@field]

Now for smaller documents (i.e. recordsets) this is not an issue, but for very large documents the overhead is enormous and totally unneccesary since the table data, which is the vast majority of the information in the document, will never contain the <structure> tag.

If the relative position of the <structure> tag never changes, or at least is known at runtime, then the for_each criteria can be changed to this...

<xsl:for-each select="/root/structure/*[name()=$zone]/row/cell[@field]">

This allows the transformation engine to go directly to the to the <structure> tag without searching the entire document. It's like the difference between giving someone specific street directions to your house or simply telling them the suburb you live in and letting them work it out from there.

The effect of this is to give a further 200% increase in speed for a total of almost a whopping 600%. Yes, that's not a typo, nearly a 6-fold improvement in xsl transformation time. The transformation times of my largest tables for 100 records are now about 12 seconds down from over 60. Needless to say, smaller recordsets are now like greased lightning

If this logic is also applied to the table_tmp variable created in my original post

<xsl:variable name="table_tmp" select="/root/*[name()=$table]" />
instead of
<xsl:variable name="table_tmp" select="//*[name()=$table]" />
then the transformation time drops again to 8 seconds. But I suspect the data structure for this

might not be so fixed, so this one is of dubious validity.

I have no idea if any of these changes have ramifications for the rest of the system, only Tony would know that, but isn't it worth investigating a 6-fold increase in speed for bugger-all effort?