Mulgara - Feature #47

(Patch) query speed improvement by 30% - 40%

02/26/2007 12:31 PM - ronald -

Status:	Closed	**Start date:**	
Priority:	High	**Due date:**	
Assignee:	ronald -	**% Done:**	0%
Category:	Mulgara	**Estimated time:**	0.00 hour
Target version:			
Resolution:	fixed		

Description

```
Creating exceptions (instances of Throwable) is an expensive
<br/>
operation, mostly because of the stack trace. However, browsing the
<br/>
code I've seen a number of places where 'new Throwable()' was used
<br/>
unconditionally. Therefore I ran some experiments.
<br/>

<br/>
Setup: a db with 1.4 million triples (production data), and 200
<br/>
&quot;production&quot; queries (most of them unique). Each test was run by
<br/>
starting mulgara, running the first 100 queries, then timing the
<br/>
second 100 queries, and finally shutting mulgara down again. The
<br/>
log level was WARN, so that normally nothing gets printed. No
<br/>
explicit transactions were used (i.e. autoCommit=true).
<br/>

<br/>
Some queries were simple (on the order of 50 ms or less to answer),
<br/>
some more complex (close to a second); the client was using SOAP to
<br/>
access the db, which introduces typically between 40 and 100 ms
<br/>
overhead per query, so these results are &quot;worst case&quot;, i.e. the
<br/>
speed-up noted is probably slightly higher using RMI.
<br/>

<br/>
Three sets of tests were run: straight rev <a href="http://mulgara.org/trac/changeset/192">192</a>
 (&quot;with stack traces&quot;),
<br/>
a version with all 'new Throwable()' removed (&quot;without stack traces&quot;),
<br/>
and a version with various 'new Throwable()' surounded by appropriate
<br/>
conditions (&quot;partial stack traces&quot;); the patch for this last set is
<br/>
the one attached to this issue.
<br/>

<br/>
Each test was run with 1, 2, and 3 client threads hitting the db.
<br/>
```

```
<br/>
Here are the results (the alignment of the table will probably be
<br/>
messed up in the html):
<br/>

<br/>
&nbsp;&nbsp;stack-traces  threads  times (sec)       avg   min    avg% min%
<br/>
&nbsp;&nbsp;
<br/>
&nbsp;&nbsp;with          1        29.2 31.4 29.0   29.9  29.0   100  100
<br/>
&nbsp;&nbsp;with          2        18.7 19.5 18.1   18.8  18.1   100  100
<br/>
&nbsp;&nbsp;with          3        17.7 17.6 19.2   18.2  17.6   100  100
<br/>
&nbsp;&nbsp;
<br/>
&nbsp;&nbsp;without       1        17.7 17.7 18.1   17.8  17.7    60   61
<br/>
&nbsp;&nbsp;without       2        13.0 12.5 13.1   12.9  12.5    69   69
<br/>
&nbsp;&nbsp;without       3        12.6 12.2 12.5   12.4  12.2    68   69
<br/>
&nbsp;&nbsp;
<br/>
&nbsp;&nbsp;partial       1        18.0 18.6 18.0   18.2  18.0    61   62
<br/>
&nbsp;&nbsp;partial       2        12.5 12.9 13.3   12.9  12.5    69   69
<br/>
&nbsp;&nbsp;partial       3        12.8 11.2 13.1   12.4  11.2    68   64
<br/>

<br/>
As you can see, the &quot;partial&quot; (which is the attached patch) gives
<br/>
around 30% - 40% reduction in query time.
<br/>

<br/>
The attached patch is quite simple and quite safe. In a couple
<br/>
cases it makes the Throwable instantiation conditional on the log
<br/>
level at which it will get printed, and in the other two cases it
<br/>
removes unnecessary (i.e. duplicate) stack-trace creation. In
<br/>
short it does not remove any information from the logs.
<br/>

<br/>
No tests are affected by this patch.
<br/>
```

History

#1 - 02/27/2007 12:34 AM - Andrae Muys -

```
I won't have time to integrate this until next week, but I'll do it first-up if Paul hasn't had a chance to ge
t to it first.
<br/>

<br/>
Nice catch - I like those numbers.
```

#2 - 03/01/2007 08:03 AM - ronald -

```
Patches applied.
```