

Re: pooled connection myth

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2005-03/1855.html>

From: John C. Bollinger (jobollin_at_indiana.edu)

Date: 03/17/05

Date: Thu, 17 Mar 2005 12:31:47 -0500

David McDivitt wrote:

- > *There is much to be gained by reusing known open connection objects, rather*
- > *than issuing single use connection objects for each request. If connection*
- > *objects are not modified by using transaction blocks, there's no reason not*
- > *to share them between threads.*

This may vary from driver to driver. In any case, however, you cannot assume that a user of the Connection will **not** use transaction blocks, so even if that were the only consideration you could not safely share Connections concurrently among threads.

- > *The word "pooling" implies this, but pooling*
- > *itself doesn't do very much.*

I'd say that "pooling" implies the opposite of sharing Connections concurrently among threads. If it were safe to share Connections in that way then there would be no need for a pool. Pooling implies need (or at least use) for multiple interchangeable instances.

- > *For the most part, all that happens is caching*
- > *the database user name and password.*

You can't have it both ways. You earlier defined what you mean by connection pooling, and very emphatically excluded simple credential caches that do not reuse Connections. I think we all agree with you that such mechanisms are not connection pools. When used appropriately pooling does what it's advertised to do, which is reduce the number of separate connections made to the database by using a few connections to support multiple connection users.

So, do some applications make use of non-pooled connections? Of course.

Do other applications make use of connection pools? Absolutely. Can junior developers be confused about whether connection pooling is in use or not? Yes. Is connection pooling useful? Sometimes.

Do you have a further point?

comp.lang.java.programmer: Re: pooled connection myth

--

John Bollinger
jobollin@indiana.edu