

Re: Synchronized Collection<T> Recommendation

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2006-04/msg04683.html>

- *From:* "Michael Primeaux" <mjprimeaux@xxxxxxx>
 - *Date:* Sun, 30 Apr 2006 15:11:31 -0500
-

William,

I think you mean to say only scalability testing (and not performance) is the only way to know for sure. Performance testing is usually done with a small number of users whereas concurrency rates are effectively measured during scalability testing. My only point is that by using lock/Monitor you prevent the consumer from deciding because you're making a statement that they have a near equal read and write frequency. Using the ReaderWriterLock class adjusts to the consumer's usage profile. For example, if the application has a higher read frequency than write frequency then—assuming an equal contention rate—the ReaderWriterLock class on a multi-CPU machine will **more than likely** result in a higher concurrency rate. If the application's read and write frequency are near identical then you might as well use lock/Monitor.

You asked what my collection does. It's as generic in usage as the Collection<T> class. The only difference is I want to return a ReaderWriterLock class for the SyncRoot property.

Kindest regards,
Michael

"William Stacey [MVP]" <william.stacey@xxxxxxxxxx> wrote in message <news:uV4qNcIbGHA.608@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

| The answer is easy; by using lock you presume to know the usage profile
| of
| the consumer is one that doesn't favor a higher read frequency. By using
| the
| ReaderWriterLock class, you favor either choice.

I think I said, don't assume anything. Let the consumer decide.

| > Is your collection doing anything besides updating a list?
| Again, by assuming an answer here you presume to know the usage profile
| of
| the consumer of the collection. In this case, you assume the to know the
| consumer favors reads equally as they favor writes. Regardless of

Re: Synchronized Collection<T> Recommendation

whether
my

I was trying not assume, that is why I asked the question. What does your collection do? As the answer can help dictate the lock to use.

ReaderWriter is not free. It is factors slower then a single monitor, so

I

would just not use it everywhere as an alternative to monitor. Plus a RW, can be unfair to writers (which may or may not be an issue for you). If your readers are processing or blocking a long time, then a RW can be a good

thing. If not, on a single cpu it will not even help. On a multi-cpu it may help a little. But if you are in and out of the read lock fast (i.e. few non-blocking instructions) then the odds are there will not be contention for the lock anyway and you only increase the odds of contention

using the slower lock (over a monitor). Naturally, testing perf is the only

way to know for sure.

—

William Stacey [MVP]