Topic: Performance characteristics / compilation affected by profiling

Replies: 1 Pages: 1 Last Post: May 3, 2006 7:29 AM by: briand

Reply to this Topic Search Forum

Back to Topic List

Replies: 1 Pages: 1

biehl

Posts: 17

Performance characteristics / compilation affected by profiling
Posted: May 3, 2006 2:33 AM

Reply

Hi,

I'm trying to do some performance critical computation in java. So I profile.

However many profilers agree on giving me a wrong profile.

1) http://profiler.netbeans.org/
2) http://jiprof.sourceforge.net/
3) http://www.yourkit.com/eap/index.jsp (in "method count" mode)

The profiles are obviously wrong as a method that is estimated to take ca. 12% of runtime really takes about 50% of the runtime (as tested by commenting the method out).

The yourkit profiler gives an answer that seems reasonable if I chose the lightest profile mode, and at the same time it seems not to list all methods (some are inlined?)

I suspect that the method counting that both 1), 2) and 3) do screws up the profiling.

Can anyone give hints to how I would construct my own profiler that gives the same measurements as the yourkit profiler in "light mode"/no method counting? Does HPROF do method counting in its polling mode?

Thanks
Anders

briand

Posts: 35

Re: Performance characteristics / compilation affected by profiling
Posted: May 3, 2006 7:29 AM

Reply

Be careful how you analyze this. When you tried to check the profiler's accuracy by commenting out the code you want to measure, the dynamics of the problem may have drastically changed. If the code was the only thing in a loop, commenting it out is likely to have caused the loop to be eliminated completely due to dead code elimation optimizations. It's difficult to tell without looking at the code in question.

Also, making a change like this is similar to eliminating a bottleneck; once eliminated, the bottleneck moves elsewhere and the profile can be dramatically different.

At least one of the three profilers you tried (the NetBeans profiler) doesn't use JVMTI for its measurements; it uses byte code instrumentation instead. HPROF, and others, use JVMTI and are likely to produce similar results. You might also want to try the Sun Studio analyzer tool:
http://developers.sun.com/prodtech/cc/index.jsp.

This tool is JVMTI based when using it to profile just the Java application. The Sun Studio Analyzer is available on Solaris and Linux.

Rather than commenting out the code, you might be better off using something like System.nanoTime() around the call and accumulating the run times and computing a percentage of the overall runtime. However, this too has certain weaknesses as System.nanoTime() has a certain amount of overhead and the system clock has a limited resolution. I would expect that this technique will return results that are similar to what the NetBeans profiler returned.

Another trick is to run with -XX:-Inline to see how the profile differs. This can sometimes shift the profile to highlight methods that were inlined into other methods.

Ultimately, though, I would hesitate to discount the results provided by the profilers. Sure, they all have certain weaknesses, but if thay are all telling you the same thing, then maybe they are all correct.

HTH
Brian

Replies: 1 Pages: 1

Back to Topic List