
Subject: Win32 UPP console application profiling? Some free easy to use tools, anyone?

Posted by [mr_ped](#) on Wed, 30 Jul 2008 19:22:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

I would love to have "profiling" directly in TheIDE, but there's no such thing.

The RTIMING (?) macro is nice when you know which part you want to attack, but I'm missing some way to measure all relevant code together and compare it in absolute/relative numbers to know which parts are slowing me down badly, down to every single function calls, preferably down to template based functions.

I was able to switch on the MSC profiling data which can be used for profiler-based compilation next time (does not interest me yet, maybe later), but the data itself are binary and I don't have idea how to view them. The file extension is pcg which is also used for some "photo CD" files, so I was unable to google anything reasonable for this one.

For GCC (MINGW) I did notice something like "gprof" mentioned on google, but it's not included with UPP MINGW, and I didn't dig further into the issue.

So, anyone has some real life experience with UPP apps profiling, and knows some free easy to use tools? Preferably for Win32, although some excellent tool for linux would made me reboot.

Subject: Re: Win32 UPP console application profiling? Some free easy to use tools, anyone?

Posted by [mirek](#) on Wed, 30 Jul 2008 20:44:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

mr_ped wrote on Wed, 30 July 2008 15:22 I would love to have "profiling" directly in TheIDE, but there's no such thing.

The RTIMING (?) macro is nice when you know which part you want to attack, but I'm missing some way to measure all relevant code together and compare it in absolute/relative numbers to know which parts are slowing me down badly, down to every single function calls, preferably down to template based functions.

I have tons of experience with RTIMING IMO it leads you directly to the bottleneck quite quickly.

Of course, if you are not afraid inserting RTIMING even to U++ code, so better. But often it is not even necessary.

It is usually quite straightforward to find out where the problem is. Just start in `CONSOLE_APP_MAIN`, if you call more than single function there. Put one RTIMING to the whole `CONSOLE_APP_MAIN`, then move other RTIMING between blocks or function calls of `MAIN`. When you identify the "clockcycle eater", descend to that block or function and repeat the process.

The main RTIMING will always serve you to see proportions...

Mirek

Subject: Re: Win32 UPP console application profiling? Some free easy to use tools, anyone?

Posted by [Novo](#) on Wed, 30 Jul 2008 22:21:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

If you consider using Linux, then combination of callgrind + kcache-grind is a perfect solution.

I've been also using Rational quantfy on Solaris, Linux, and Windows. It works well, and it costs a lot.

gprof just doesn't work.

Subject: Re: Win32 UPP console application profiling? Some free easy to use tools, anyone?

Posted by [mr_ped](#) on Thu, 31 Jul 2008 07:56:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

Mirek: that's exactly what will not work for me. I have one big processing functions which eats 100% of CPU, and I know why/how, and there's no direct bottleneck in it (or other way around, everything in it is bottleneck). So I'm not looking into searching for bottleneck, but to tune little parts one by one, and to know where to start I would like overall view.

Novo: thanks for advice, I'm sort of looking at valgrind(+proper modules) too, but unfortunately right now I don't have linux distribution at my notebook (and last time I tried to install my favorite Kubuntu it had some problems with HW), where I spend half of my development time, so I will check it later.

Anyway, I did find something called "Shiny" <http://sourceforge.net/projects/shinyprofiler> (it has similar license to U++), which does require putting something into your code (so it very likely breaks inlining and other optimizations), so it's similar to RTIMING, yet there's the ready to use `std::ostream` text output, so it does save you some "LOG << this is result" work when compared to RTIMING.

Now I see I did download official zip which is a bit behind current SVN, but after tiny fixes/changes it works for me, and I don't want to spend more time with it now.

I'm adding my UPP packages if anyone is interested (I did put them into `upp/bazaar` at my HDD, as I'm using the UnitTest++ package from it too). But I don't think it is ready to be included with UPP in next versions.

If anyone has spare time and is interested, feel free to examine this package, update it from SVN and make sure it works under many platforms, the license is friendly, so eventually we may add it to bazaar in future(?).

As is, it works for me with 2008.1 under WinXP with both MSC8 and MINGW.

File Attachments

1) [ShinyUPP.7z](#), downloaded 397 times

Subject: Re: Win32 UPP console application profiling? Some free easy to use tools, anyone?

Posted by [mr_ped](#) on Thu, 31 Jul 2008 07:58:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

One thing which I don't like about it right now is, that under MSC I get in output full "MyClass::Get", but under MINGW I get just "Get", which is of course confusing a lot when you have "Get" in several classes.

If anyone would be able to fix this at Shiny sources, I would thank him a lot.

Subject: Re: Win32 UPP console application profiling? Some free easy to use tools, anyone?

Posted by [mirek](#) on Thu, 31 Jul 2008 08:31:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

mr_ped wrote on Thu, 31 July 2008 03:56it later.

Anyway, I did find something called "Shiny" <http://sourceforge.net/projects/shinyprofiler> (it has similar license to U++), which does require putting something into your code (so it very likely breaks inlining and other optimizations), so it's similar to RTIMING, yet there's the ready to use std::ostream text output, so it does save you some "LOG << this is result" work when compared to RTIMING.

Ehm, maybe you should check reference example related to TIMING. Of course, there are no more LOGs with it.

Just use

```
{
  RTIMING("MyBlockTiming");
  ....
}
```

and that is all.

Mirek

Subject: Re: Win32 UPP console application profiling? Some free easy to use tools, anyone?

Posted by [mr_ped](#) on Thu, 31 Jul 2008 08:34:29 GMT

[View Forum Message](#) <> [Reply to Message](#)

hmm... maybe I should check reference next time, you are right.
(still the Shiny has the call-tree too, so it has some added value anyway)

edit: I tried your RTIMING way, and honestly, the Shiny output gives much better overview. Although for single routine tuning the RTIMING is perfectly valid solution, and built-in into UPP Core, thanks for the "tutorial", now I will know.
