Subject: Re: Ready to quit after 25 years...
Posted by Haje Korth on Tue, 12 Jul 2011 12:16:06 GMT
View Forum Message <> Reply to Message

Chris,
thanks for the insight on the inner workings of profiler, and your
change makes a lot of sense to not lead the user down the wrong path.
I look forward to the vectorized arrow function to get 2+ order
magnitude speed improvement. I volunteer as beta tester. :-)

Haje

On Jul 8, 12:53 pm, Chris Torrence <gorth...@gmail.com> wrote:
> Hi Bob,
>
> Thanks for the kind remarks!
>
> Now, regarding the problem with Arg_Present - it turns out to not be a problem with Arg_Present at all. The Profiler introduces a slight overhead when it computes the time spent per iteration and records the information in an internal data structure. For most routines this overhead is negligible compared to the execution time of the routine. However, for very fast routines (like Arg_Present), the overhead is significant. The real problem is that Arg_Present was getting called 15 million times. So what you were really measuring for things like Arg_Present, Obj_Isa, etc. is just the speed of the profiler!
>
> I just made a change to the way the Profiler works. Now, by default it will not include the following routines: ARG_PRESENT, KEYWORD_SET, N_ELEMENTS, OBJ_ISA, OBJ_VALID, PTR_VALID, SIZE. You can still turn on profiling for these routines, but you need to do it explicitly. For example:
> PROFILER, /SYSTEM ; turns on profiling for all system routines except the above
> PROFILER, 'ARG_PRESENT', /SYSTEM ; also turns it on
>
> Hopefully with this change, your profiling results will no longer be biased towards routines like Arg_Present and N_Elements, and you will be better able to discover the real bottlenecks in your code.
>
> For example, in Haje's case with the Arrow function, the real problem is that he has to call Arrow 1500 times because it doesn't accept array arguments. We're also working on that for IDL 8.2.
>
> Cheers,
> Chris
> ITTVIS