Subject: Re: IDL and OpenGL
Posted by ushomirs on Wed, 29 Sep 1999 07:00:00 GMT
View Forum Message <> Reply to Message

I think what SGI folks meant is that IDL routines are **not** compiled into **machine** code, the way C programs are. So even an IDL routine is "compiled", it is transformed into a bytecode for it's own virtual machine. So unlike compiled C code, where the processor executes your program directly, the processor executes the code for the "IDL virtual machine", which in turn reads your program (or the "compiled" bytecode) and interprets it step by step. So there is an extra level of indirection involved in running IDL code. that's what the SGI people were refering to.

If IDL code were truly compiled in the same sense as C code, then this loop
FOR i=0,9 do foo[i]=2.*foo[i]

would take exactly as long as

foo=2.*foo

It doesn't, because the IDL "virtual machine" has to interpret every iteration of the FOR loop, but it can just map the vector operation onto a machine code for loop.

Now, in case of OpenGL, this distinction is a wash. So long as most of the time is spent inside OpenGL, some slowness associated with interpreting rather than executing machine code directly doesn't matter.

greg

THere's some discussion of this in IDL's external development guide.

In this respect IDL is much more like Java, where th

In article <37F22743.52AC5811@ssec.wisc.edu>,
Liam Gumley <Liam.Gumley@ssec.wisc.edu> wrote:
> Richard Tyc wrote:
>> Our group was having a meeting with some SGI folks yesterday and some
>> interesting points were brought up which I hope some IDL experts could shed

>> some light on. We were discussing the performance of IDL vs. OpenGL source
>> based app.. Their point was that IDL is an interpreted language and as such
>> is not optimized (or tuned as they put it) for OpenGL and thus runs
>> significantly slower than a custom C app using OpenGL.
>
> IDL does indeed run in an interpreting mode (at the command line), where
> each statement is interpreted and executed separately. However once you
> 'compile' an IDL procedure, it is no longer 'interpreted' each time it
> is called.
>
> Perhaps someone in the know could give us a brief description of how IDL
> procedures are transformed from source code to CPU instructions.
>
> Cheers,
> Liam.
>
> --
> Liam E. Gumley
> Space Science and Engineering Center, UW-Madison
> http://cimss.ssec.wisc.edu/~gumley
>