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

ushomirs@my-deja.com wrote in message <7stme3$g15$1@nnrp1.deja.com>...
> 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

So are you saying when I am doing things like oWindow->Draw, oView and the
scene is full of 3D objects, Light objects, texture mapped images etc. IDL
should perform pretty well because this mostly happens in hardware via
OpenGL lib calls (this is what I am hoping ? Upgraded graphics hardware
should meet our needs then - things like geometry and rasterization accel in
hardware chipsets etc.) But when it comes to pure number crunching in the
app, it may run a bit slower ?

Thanks for the info. I am trying to understand the limitations of IDL.
Right now, we are running on fairly slow hardware (SGI O2 R5k) and I have
unacceptable 3D performance when it comes to realtime updates of Views with
complex volume objects being rendered.   It's far easier to sink $10-15K
down on faster hardware (dual CPU to benefit the IDLgrVolume object) then

start looking at a complete overhaul of our app using a different
development environment.

Rich