Subject: Re: IDL and OpenGL
Posted by Karl Schultz on Fri, 08 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Richard Tyc wrote in message <7stqebjmk1@canopus.cc.umanitoba.ca>...

> 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 ?

The IDL interpreter simply calls some compiled and optimized C code that implements the Draw method. This code takes the graphic data associated with the view and draws it with OpenGL as fast as it can manage. At this point, the performance is bound mostly by the OpenGL implementation and any hardware beneath it, since the view's graphic data is "ready for submission" to the graphics system. The IDL interpeter gets control after the Draw() method completes. So, yes, any improvement in graphics hardware should improve graphics performance by about the same factor. This really only applies if you don't pop back into IDL to do a lot of recalculation for each graphic "frame".

If you have hardware-accelerated graphics, you may also see a net improvement over the total time for running the app since the graphics processors can offload the main CPU, giving it more time for other work. This would especially be true of graphics systems that implement the transformation pipeline in hardware.

> 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.

So true.