Subject: alternative to execute
Posted by greg michael on Tue, 31 Jan 2006 18:43:04 GMT
View Forum Message <> Reply to Message

I'm trying to think up a good way to implement a user-defined function which would run in the IDL VM. The function might involve any of about 300 similar 2-d arrays, each of which is quite expensive to generate. So I thought to obtain the function as a string, scan it to see which arrays are needed, generate them, and then apply the function with 'execute'. But 'execute' is excluded from the VM.

A typical function might look like:

result = b123 / b100 - .9 * b050 / b035

Without execute, I can only think to attempt to write some kind of arithmetic interpreter. Would anyone have a suggestion for a way to get IDL to do this more directly?

Subject: Re: alternative to execute
Posted by JD Smith on Wed, 01 Feb 2006 21:17:56 GMT
View Forum Message <> Reply to Message

On Wed, 01 Feb 2006 03:04:12 -0800, greg michael wrote:

> Thanks for this exotic solution, but I'm afraid, as Marc says, the file
> access would be a problem.

File access isn't a problem under the VM, but if the particular file you are trying to access is a .pro file, and the particular thing you are trying to do is compile that file, then you're out of luck. For obvious reasons, the IDL VM provides no access to the IDL byte-compiler. As long as the set of arguments is not too complex, writing a simple stack-based arithmetic parser in IDL is not too bad, and probably the only solution.

JD

Subject: Re: alternative to execute
Posted by greg michael on Wed, 01 Feb 2006 22:38:29 GMT
View Forum Message <> Reply to Message

Thanks, JD - yes, I think that's what I shall do.

Subject: Re: alternative to execute
Posted by news.qwest.net on Wed, 01 Feb 2006 23:03:57 GMT
View Forum Message <> Reply to Message

"greg michael" <greg.michael@gmail.com> wrote in message
news:1138732984.579226.255440@f14g2000cwb.googlegroups.com.. .
>
> I'm trying to think up a good way to implement a user-defined function
> which would run in the IDL VM. The function might involve any of about
> 300 similar 2-d arrays, each of which is quite expensive to generate.
> So I thought to obtain the function as a string, scan it to see which
> arrays are needed, generate them, and then apply the function with
> 'execute'. But 'execute' is excluded from the VM.
>
> A typical function might look like:
>
> result = b123 / b100 - .9 * b050 / b035
>
> Without execute, I can only think to attempt to write some kind of
> arithmetic interpreter. Would anyone have a suggestion for a way to get
> IDL to do this more directly?
>

From your example equations, it seems that all you want is a calculator
that operates on arbitrary arrays. So, my first impulse would be to
greatly restrict the number of functions.
You don't need to have a function for every possible expression.

How about a function for addition, subtraction, multiplication and division,
each accepting 2 inputs and returning one output.
Of course, you can add some simple functions like sqrt to it easy enough.

Then save the result as a new array, and allow further operations on it.
If you had a "workspace" (like 'memory' on a calculator) where the user
could save their results, then it is really easy to make the full
expression.
Each term in a complex expression becomes one entry in the memory,
then you add them all together to get the final result.

Cheers,
bob

Subject: Re: alternative to execute
Posted by greg michael on Thu, 02 Feb 2006 11:58:55 GMT
View Forum Message <> Reply to Message

Hi Bob,
I'm writing a program to process a spectral image cube on the fly. Once defined, the function would be applied to whichever part of the cube is being viewed. So yes, it's like a calculator, but I don't want to use a workspace because it needs to work without any further interaction. But your suggestion to convert operators to functions could simplify the parser - I'll think about that.
many thanks,
Greg

Subject: Re: alternative to execute
Posted by news.qwest.net on Thu, 02 Feb 2006 17:28:06 GMT
View Forum Message <> Reply to Message

"greg michael" <greg.michael@gmail.com> wrote in message
news:1138881535.246386.5040@f14g2000cwb.googlegroups.com...
> Hi Bob,
> I'm writing a program to process a spectral image cube on the fly. Once
> defined, the function would be applied to whichever part of the cube is
> being viewed. So yes, it's like a calculator, but I don't want to use a
> workspace because it needs to work without any further interaction. But
> your suggestion to convert operators to functions could simplify the
> parser - I'll think about that.
> many thanks,
> Greg

I see. The user defined function (udf) will be applied only to a subset of the
overall data, then the user will scroll around necessitating that the function
be re-applied (is that right). So, it is not possible to perform the operation
on the entire spectral image cube? (memory requirements?).

On the other hand, it seems like you want to give the user unlimited power
in creating a function, but is that really necessary? That basically requires that you
write your own language or operating system.(with "execute()" you would be leveraging
IDLs language, but with out it, you have to create your own).

Perhaps you can get 99% of the functionality if you create a general function,
and allow the user to input the parameters. For matrices m1,m2,m3,m4 you could have
a_1*(m1)^b_1 + a_2*(m2)^b_2+....

Then just have a series of boxes where the user inputs the parameters
a_1,a_2,etc and b_1,b_2,etc,
as well as the input matrices m1, m2...

It would be a bit more complicated if you want to do operations with the
matrices on other
matrices, but you could make a couple general functions to cover most cases

[a_1*(m1)^b_1]*[a_2*(m2)^b_2]

where of course b_2 = -1 to make it a division.

So there you have 2 general functions, and that should cover a great deal of
the user defined functions. Of course you can expand this according to your
actual needs.

Cheers,
bob

PS another idea, can you create a C language DLL or a DLM that could perform
a similar
function to the "execute" command? I don't know, just an idle thought.

Subject: Re: alternative to execute
Posted by greg michael on Thu, 02 Feb 2006 18:46:47 GMT
View Forum Message <> Reply to Message

Hi Bob,

> I see. The user defined function (udf) will be applied only to a subset of the
> overall data, then the user will scroll around necessitating that the function
> be re-applied (is that right). So, it is not possible to perform the operation
> on the entire spectral image cube? (memory requirements?).

Exactly. I want to avoid processing the entire image for speed.

For the moment, I'm prototyping it with execute (which works well), but
I'm thinking now to attempt an arithmetic interpreter. I imagine some
nice recursive routine with regex string cutting (I dread that part!)
and cases for each operator could do it. I'm not keen to use general
functions for the loss of flexibility, although you're right - if well
chosen, they could probably cover most cases.

As for C - haven't touched that in years - wouldn't that need some kind
of self-compiling code? Sounds intimidating to me!

thanks again for your comments - much appreciated.

Greg