Subject: call external speed
Posted by curtis volin on Wed, 02 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I use the IDL SVD command very frequently, so I was wondering if it would
be any faster to create a DLL and call it with call external. I suppose
that the fundamental question is, are IDL routines optimized for speed?

Thanks,

Curtis Volin
cev@u.arizona.edu

Research Associate
Optical Sciences Center
University of Arizona

Subject: Re: call external speed
Posted by Karl Krieger on Thu, 03 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

On Wed, 2 Sep 1998, curtis volin wrote:

> I use the IDL SVD command very frequently, so I was wondering if it would
> be any faster to create a DLL and call it with call external. I suppose
> that the fundamental question is, are IDL routines optimized for speed?

I guess it depends on the algorithm. A function designed for a specific purpose will often be significantly faster if you code it with a compiling language and link it to IDL by call_external. Just run a test case and compare IDL's SVD to an implementation of SVD in C or Fortran.

BTW: I would recommend to use LINKIMAGE instead of CALL_EXTERNAL except perhaps for some quick and dirty hacks. With LINKIMAGE your procedures and functions have exactly the same syntax than the rest of IDL; you get argument checking, proper IDL error handling and it is possible to pass keyword parameters and structures to your routine. The downside is that it requires some more programming effort but it's really worth it IMHO.

Karl
--
Max-Planck-Institute for Plasma Physics
Boltzmannstr.2, 85740 Garching, Germany   Email: krieger@ipp.mpg.de

Subject: Re: call external speed

Posted by steinhh on Fri, 04 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

In article <6soch3$5da$1@nnrp1.dejanews.com>
rmlongfield@my-dejanews.com writes:

> I have a question though: Which chapters in the Advanced Development guide
> are related to LINKIMAGE? There seems to be a mixture of topics. Do you have
> a short example?   I can't find any in the IDL documentation. (I work with
> UNIX on an SGI workstation.)

Now I don't have the printed documentation, but the online help does have some examples (IDLv5.1). Under the "Contents" in the normal online help, double-click on the *book* "IDL External Development Guide", double-click on "Open IDL Ext....". Then press the Contents button, and double-click again on "External Development Guide" - a list of chapters should appear.

"Adding system routines" is crucial to LINKIMAGE. Most of the other chapters are relevant, too, depending on what you want to do in your routines.

The Caution under LINKIMAGE is appropriate, but only to a certain extent. You do *not* have to have "intimate knowledge of the internals of IDL" in the sense that you have to qualify for work in RSI's development group - not by a long shot. The very least you have to know about is how variables are *stored*, *passed*, and returned. That's basically it (for writing something simple that works - for the right reasons).

Using Callable IDL has nothing as such to do with LINKIMAGE versus CALL_EXTERNAL, all permutations are possible.

However, you *have* to use Callable IDL to install new system procedures from inside CALL_EXTERNAL routines (i.e., *not* using LINKIMAGE, but the IDL_AddSystemRoutine function).

On the other hand, if you're using LINKIMAGE, Callable IDL provides some *very* useful routines/macros to do checks on your input data etc. For example, IDL_ENSURE_ARRAY will issue standard "system" error messages and associated behaviour (stopping etc), in exactly the same way that e.g. MEDIAN does:

```
IDL> print,median(0)
% MEDIAN: Expression must be an array in this context: <INT      (       0)>.
% Execution halted at:  $MAIN$
```

Callable IDL also provides routines for keyword processing, type conversion etc. *Very* handy.

One thing that I cannot find in the online help is a good
listing of "these routines and macros are available when
linking your shareable objects with the idl library
(i.e., using Callable IDL)". There is a list for RPC
function calls, but not Callable IDL. Thus, you have to
guess, and read through many chapters and a lot of text
to find what you need (or just drop the whole thing
because you don't know what you're missing!).

Regards,

Stein Vidar

Subject: Re: call external speed
Posted by rmlongfield on Fri, 04 Sep 1998 07:00:00 GMT

View Forum Message <> Reply to Message

In article <Pine.SO4.4.02.9809031131320.2992-100000@sukak>,
Karl Krieger <kak@ipp.mpg.de> wrote:

> BTW: I would recommend to use LINKIMAGE instead of CALL_EXTERNAL except
> perhaps for some quick and dirty hacks. With LINKIMAGE your procedures and
> functions have exactly the same syntax than the rest of IDL; you get
> argument checking, proper IDL error handling and it is possible to pass
> keyword parameters and structures to your routine. The downside is that it
> requires some more programming effort but it's really worth it IMHO.

Hi Karl, I've just figured out how to use CALL_EXTERNAL but am curious
about LINKIMAGE. The documentation is very sparse and the "Caution" is
discouraging. My main goal is to leave my C programs intact and independent
of IDL, i.e. I do not want to use callable IDL.

I have a question though: Which chapters in the Advanced Development guide
are related to LINKIMAGE? There seems to be a mixture of topics. Do you have
a short example? I can't find any in the IDL documentation. (I work with
UNIX on an SGI workstation.)

Most likely I will stay with CALL_EXTERNAL for now, since it is working.
But maybe in the future I will look into LINKIMAGE.

Rose

Subject: Re: call external speed
Posted by David Foster on Tue, 08 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

curtis volin wrote:
>
> I use the IDL SVD command very frequently, so I was wondering if it would
> be any faster to create a DLL and call it with call external. I suppose
> that the fundamental question is, are IDL routines optimized for speed?
>

Curtis -

Once I had to write a specialized interpolation algorithm, so I coded
it in C and called it using CALL_EXTERNAL. I made my routine as general
as I could, so performing the standard interpolation would be an
option. Just for kicks, I decided to compare the speed of my
routine and IDL's INTERPOLATE() routine. The results were *very*
similar. I would say that it is fairly safe to say that IDL's
routines are well optimized.

Dave
--

~~~~~~ ~~~~~~~
David S. Foster Univ. of California, San Diego
Programmer/Analyst Brain Image Analysis Laboratory
foster@bial1.ucsd.edu Department of Psychiatry
(619) 622-5892 8950 Via La Jolla Drive, Suite 2240
La Jolla, CA 92037
~~~~~~ ~~~~~~~

Subject: Re: call external speed
Posted by Karl Krieger on Mon, 14 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

On Tue, 8 Sep 1998, David Foster wrote:

> ... Just for kicks, I decided to compare the speed of my
> routine and IDL's INTERPOLATE() routine. The results were *very*
> similar. I would say that it is fairly safe to say that IDL's
> routines are well optimized.

It really depends on the application. I wrote a LINKIMAGE wrapper for the
FFTW package (http://theory.lcs.mit.edu/~fftw) and compared the speed to
IDL's native FFT routine. The speed gain for single precision
real->complex 2d transforms is about 2.5 on a SUN Ultra/170 and about 2.3

on a Pentium/133 under WinNT, so it's really worth the effort if you want to do FFT of large data sets.

Karl
--
Max-Planck-Institute for Plasma Physics
Boltzmannstr.2, 85740 Garching, Germany Email: krieger@ipp.mpg.de