Subject: Re: Zero... THANKS Pavel and J.D.
Posted by davidf on Thu, 01 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Frank Morgan (frank.morgan@jhuapl.edu) writes:

> Incedentally, I'd never realized you can say (where())[0] to get that first
> element - that's a handy statement.

Handy, but really, really dangerous. I wouldn't use it in
code, personally, unless I had bomb-proof error handling
in place. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Zero... THANKS Pavel and J.D.
Posted by Frank Morgan on Thu, 01 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Pavel and J.D.,

Thanks to both of you for some useful information.

Pavel mentioned the 0-at-the-end problem with the loop approach. After some analysis of my problem (a directed graph search), I've discovered that most of the vectors I search will contain a non-zero within the first 10% or so of the length. I would have thought then that the loop would be faster than 'where' but J.D.'s timing results prove me wrong - looks like even with loop halt at 10% (where his test halted at 50%), the timing would be 0.99/5 = 0.2, still twice the time of 'where' searching the whole vector - boy, IDL loops really are bad!

For now I'm settling on 'where' - it's just fast enough for the biggest graphs I'm searching so far. J.D.'s timing for external code indicates that with 10% lengths typical, I might get 10X speedup over 'where' but for now the DLL compilation isn't worth it. But it gives me an out if I need to search bigger graphs.

Incedentally, I'd never realized you can say (where())[0] to get that first
element - that's a handy statement.

Thaks,
Frank

Subject: Re: Zero... THANKS Pavel and J.D.
Posted by R.Bauer on Fri, 02 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Frank Morgan wrote:

> Pavel and J.D.,
>
> Thanks to both of you for some useful information.
>
> Pavel mentioned the 0-at-the-end problem with the loop approach. After some
> analysis of my problem (a directed graph search), I've discovered that most
> of the vectors I search will contain a non-zero within the first 10% or so
> of the length. I would have thought then that the loop would be faster
> than 'where' but J.D.'s timing results prove me wrong - looks like even with
> loop halt at 10% (where his test halted at 50%), the timing would be 0.99/5
> = 0.2, still twice the time of 'where' searching the whole vector - boy, IDL
> loops really are bad!
>
> For now I'm settling on 'where' - it's just fast enough for the biggest
> graphs I'm searching so far. J.D.'s timing for external code indicates that
> with 10% lengths typical, I might get 10X speedup over 'where' but for now
> the DLL compilation isn't worth it. But it gives me an out if I need to
> search bigger graphs.
>
> Incedentally, I'd never realized you can say (where())[0] to get that first
> element - that's a handy statement.

It is always better to use the count value.
 where(a eq 0 ,count)

if count gt 0 then ...

R.Bauer

>
>
> Thaks,
> Frank
