Subject: Re: speed of n_elements
Posted by Craig Markwardt on Wed, 03 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

>

- >> * I can't repeat your experience on Sun or Linux. On both those
- >> machines n_elements(data.flag) is much slower than n_elements(data),
- >> at least in a loop.

>

- program deleted ...
- > IDL> a=fltarr(1000)
- > IDL> b={a:a, b:a, c:a, d:a, flag:0L, name:", other:0.0}
- > IDL> c=replicate(b, 500)
- > % Compiled module: TEST.
- > IDL> k={a:0L, data:c}
- > IDL> test,k
- > 33.133333
- > 0.0333333302

Okay, now I agree with you. It should be clear that branch that takes longer, a.data, has to extract a lot more data than a.data.flag. As I said some reason IDL is very clumsy about structures with big arrays in them (and I was talking about megabyte arrays).

- > ... In my opinion, the only drawback in
- > using pointers and (built-in) objects is that they take away from you some of
- > the tremendous advantages that IDL has in handling arrays.

I agree. I'm not a huge fan of pointers unless I have to, especially since I'm still trying to keep some IDL 4 compatibility.

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: speed of n_elements
Posted by Pavel Romashkin on Wed, 03 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

- > * I can't repeat your experience on Sun or Linux. On both those
- > machines n_elements(data.flag) is much slower than n_elements(data),
- > at least in a loop.

Thank you for the test, Craig. I repeated it and n_elements(data) was way faster. But I was certain that I saw what i saw, so I reconstructed the entire data system in the same way it was in my program. DATA in my program is itself passed as a field of State structure inside a widget tree. Try the following code:

```
pro test, a
start=systime(1)
for i=0, 100 do temp = n elements(a.data)
print, systime(1)-start
start=systime(1)
for i=0, 100 do temp = n_elements(a.data.flag)
print, systime(1)-start
end
****** ; Now, the outputs:
IDL> a=fltarr(1000)
IDL> b={a:a, b:a, c:a, d:a, flag:0L, name:", other:0.0}
IDL> c=replicate(b, 500)
% Compiled module: TEST.
IDL> k={a:0L, data:c}
IDL> test,k
    33.133333
  0.033333302
```

- > * I've always found that putting large data arrays into structures is
- > a big loser. In my experience it's slow to create such structures
- > and slow to extract the fields later.

I agree it is not nearly as neat as pointers. I don't find it noticeably slow, at least with my array sizes. However, there are things that are really advantageous in placing arrays directly as structure fields. For instance, I can shift all arrays in the array of structures in one line:

```
data.(k) = shift(data.(k), n_pts)
```

while with pointers this does not work. In my opinion, the only drawback in using pointers and (built-in) objects is that they take away from you some of the tremendous advantages that IDL has in handling arrays.

Cheers, Pavel

P.S. Not to mention the humiliating use of HEAP_GC after a crash of an app

Subject: Re: speed of n_elements
Posted by Craig Markwardt on Wed, 03 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

- > So, briefly: if you want to get the size of an array of structures,
- > examine the size of an array of scalar fields (if available) and you
- > will have 10X5 faster n_elements.

Two things:

- * I can't repeat your experience on Sun or Linux. On both those machines n_elements(data.flag) is much slower than n_elements(data), at least in a loop.
- * I've always found that putting large data arrays into structures is a big loser. In my experience it's slow to create such structures and slow to extract the fields later. Thus I was surprised by your observation (but unfortunately I can't confirm it); however because of that I still avoid large arrays in structures. Use pointers or handles instead.

Craig	
,	craigmnet@cow.physics.wisc.edu Remove "net" for better response

Subject: Re: speed of n_elements
Posted by R.Bauer on Thu, 04 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Pavel Romashkin wrote:

- >> * I can't repeat your experience on Sun or Linux. On both those
- >> machines n_elements(data.flag) is much slower than n_elements(data),
- >> at least in a loop.
- > Thank you for the test, Craig. I repeated it and n_elements(data) was way
- > faster. But I was certain that I saw what i saw, so I reconstructed the
- > entire data system in the same way it was in my program. DATA in my program

```
> is itself passed as a field of State structure inside a widget tree. Try the
  following code:
  pro test, a
>
> start=systime(1)
> for i=0, 100 do temp = n_elements(a.data)
  print, systime(1)-start
> start=systime(1)
> for i=0, 100 do temp = n_elements(a.data.flag)
> print, systime(1)-start
>
> end
Hi
This example is 100 times faster.
R.Bauer
pro test, a
start=systime(1)
c=a.data; changed!!
for i=0, 100 do temp = n_elements(c)
print, systime(1)-start
start=systime(1)
for i=0, 100 do temp = n_elements(a.data.flag)
print, systime(1)-start
end
```