Subject: Re: Structure of arrays or arrays of structures?
Posted by Michael Galloy on Thu, 22 Jan 2009 22:32:03 GMT
View Forum Message <> Reply to Message

On Jan 22, 1:52 pm, Brian Larsen <balar...@gmail.com> wrote:

> I prefer the feel of the structure of arrays since I like typing
> tmp = data.jd[0:10]
> more than I like typing
> tmp = data[0:10].jd

You can use either notation with an array of structures:

```
IDL> s = replicate({ a: 0, b: 0 }, 10)
IDL> s.a = findgen(10)
IDL> s.b = 3
IDL> print, s.a
       0       1       2       3       4       5       6       7
8       9
IDL> print, s.b
       3       3       3       3       3       3       3       3
3       3
IDL> print, s[7]
{       7       3}
```

Mike
--
www.michaelgalloy.com
Tech-X Corporation
Associate Research Scientist

Subject: Re: Structure of arrays or arrays of structures?
Posted by TonyL on Thu, 22 Jan 2009 22:57:46 GMT
View Forum Message <> Reply to Message

I think the array of structures is a preferable method because it groups sets of data into consistent structures. When i store variables into sav files for later deployment, i like to see where the sets occur when i restore them.

Tony

Subject: Re: Structure of arrays or arrays of structures?
Posted by Craig Markwardt on Fri, 23 Jan 2009 00:15:06 GMT
View Forum Message <> Reply to Message

On Jan 22, 3:52 pm, Brian Larsen <balar...@gmail.com> wrote:

```
> All,
>
> this may be philosophy but I have fought with both today and I am
> wondering if there are pros and cons to the different
> implementations.
>
> My example is that when I read in text files I build a structure of
> arrays:
> IDL> help, data
> DATA            STRUCT    = -> <Anonymous> Array[1]
> IDL> help, data, /str
> ** Structure <1a07a08>, 5 tags, length=212480, data length=212480,
> refs=1:
>    JD              DOUBLE    Array[5312]
>    PAS0            DOUBLE    Array[5312]
>    PAS90           DOUBLE    Array[5312]
>    MEP0E3          DOUBLE    Array[5312]
>    MEP90E3         DOUBLE    Array[5312]
>
> and when I read cdf data (at least from ACE) I get arrays of
> structures
> IDL> help, data
> DATA            STRUCT    = -> <Anonymous> Array[36451]
> IDL> help, data, /str
> ** Structure <19cf008>, 45 tags, length=200, data length=196, refs=1:
>    DNUM            DOUBLE           0.0000000
>    YEAR            LONG              2007
>    DAY             LONG               165
>    HR              LONG                 0
>    MIN             LONG                 0
>    SEC             FLOAT           8.98560
>
> Are there memory issues with one way or the other? Other things I
> haven't thought about?
>
> I prefer the feel of the structure of arrays since I like typing
>    tmp = data.jd[0:10]
> more than I like typing
>    tmp = data[0:10].jd
```

My thoughts:

Both methods will be slower than pure arrays.

I believe the array-of-structures approach will be the slowest,
because every time you need to fetch a field (as your second example),
IDL has to assemble the array from structure parts. That's been my

experience anyway.

However, the array-of-structures approach is natural if you can think
of your data as a database, and you can add or remove more entries at
a later time. You can do this easily by filtering the array of
structures and/or appending, without destroying the existing entries.
For a structure of arrays you basically need to tear apart the
structure and re-build it if you want to change any of the contents.

The structure-of-arrays approach is natural when you will be passing
it as an _EXTRA argument, since for that you need all of your data
bundled into a single structure.

Craig

Subject: Re: Structure of arrays or arrays of structures?
Posted by R.Bauer on Fri, 23 Jan 2009 08:11:36 GMT

View Forum Message <> Reply to Message

Brian Larsen schrieb:

```
> All,
>
> this may be philosophy but I have fought with both today and I am
> wondering if there are pros and cons to the different
> implementations.
>
> My example is that when I read in text files I build a structure of
> arrays:
> IDL> help, data
> DATA            STRUCT    = -> <Anonymous> Array[1]
> IDL> help, data, /str
> ** Structure <1a07a08>, 5 tags, length=212480, data length=212480,
> refs=1:
>    JD              DOUBLE    Array[5312]
>    PAS0            DOUBLE    Array[5312]
>    PAS90           DOUBLE    Array[5312]
>    MEP0E3          DOUBLE    Array[5312]
>    MEP90E3         DOUBLE    Array[5312]
>
> and when I read cdf data (at least from ACE) I get arrays of
> structures
> IDL> help, data
> DATA            STRUCT    = -> <Anonymous> Array[36451]
> IDL> help, data, /str
> ** Structure <19cf008>, 45 tags, length=200, data length=196, refs=1:
>    DNUM            DOUBLE          0.0000000
>    YEAR            LONG            2007
>    DAY           LONG            165
>    HR            LONG              0
>    MIN           LONG              0
>    SEC           FLOAT       8.98560
```

>

> Are there memory issues with one way or the other? Other things I
> haven't thought about?

>

> I prefer the feel of the structure of arrays since I like typing
>   tmp = data.jd[0:10]
> more than I like typing
>   tmp = data[0:10].jd

Hi

well with the last one you can quite easy create slices of your data set
by result = data[ix]. It also could be used with table widgets or for an
example by xvaredit.

Your prefered solution is easy to use for all kind of tag manipulation,
e.g. removing, renaming, adding another tag into the same structure.

Long time ago we have added reform_struct to our library.
http://www.fz-juelich.de/zb/datapool/page/439/00322_Bauer.pd f page 57
(text in german, examples in idl)

cheers
Reimar

>
>
> Cheers,
>
> Brian
>
> -- --------------
> Brian Larsen
> Boston University
> Center for Space Physics
> http://people.bu.edu/balarsen/Home/IDL

Subject: Re: Structure of arrays or arrays of structures?

Posted by Brian Larsen on Fri, 23 Jan 2009 15:23:31 GMT
View Forum Message <> Reply to Message

Thanks all,

Some good points were raised, the statements about "...can quite easy create slices of your data set" and "...groups sets of data into consistent structures" resonated with me. I will play with array of structures for a while and see what I think. I have a feeling it will remedy a lot of annoying where() statements that occur since you cant "drop" array elements from a structure of arrays.

Brian

 --- --------------
Brian Larsen
Boston University
Center for Space Physics
http://people.bu.edu/balarsen/Home/IDL

Subject: Re: Structure of arrays or arrays of structures?
Posted by Paul Van Delst[1] on Fri, 23 Jan 2009 17:25:09 GMT
View Forum Message <> Reply to Message

Brian Larsen wrote:
> Thanks all,
>
> Some good points were raised, the statements about "...can quite easy
> create slices of your data set" and "...groups sets of data into
> consistent structures" resonated with me. I will play with array of
> structures for a while and see what I think. I have a feeling it will
> remedy a lot of annoying where() statements that occur since you cant
> "drop" array elements from a structure of arrays.

My $0.02: I reckon it depends on the problem you're dealing with, or how your data is assembled, or how you interact with your data. IMO, neither approach is suitable as an "always do it this way" method.

cheers,

paulv

Subject: Re: Structure of arrays or arrays of structures?
Posted by David Fanning on Fri, 23 Jan 2009 17:27:53 GMT

Paul van Delst writes:

> My $0.02: I reckon it depends on the problem you're dealing with, or how your data is
> assembled, or how you interact with your data. IMO, neither approach is suitable as an
> "always do it this way" method.

Wow! I guess there *has* been a change in Washington! ;-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")