
Subject: Re: use of temporary in IDL

Posted by [Dick Jackson](#) on Thu, 08 Nov 2012 02:37:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

Rohit,

Indexing using the `[*, *, sorted_z]` creates a list of indices with a 4-byte value for each element, leading to that huge memory usage. In this case, making a second copy (perhaps unavoidable), then looping over one dimension ends up using far less memory and taking less time. (Sometimes a loop is OK, especially when one statement still operates on 4 million elements!):

PRO IndexingTest

```
nPlanes = 287
```

```
sorted_z = Sort(RandomU(42L, nPlanes))
```

```
vol = IntArr( 512, 512, nPlanes )
```

```
vol[0, 0, *] = IndGen(nPlanes) ; Put [0,1,2...] in position [0,0] on each plane
```

```
mem0 = (Memory())[0] ; Current memory usage
```

```
t0 = SysTime(/Seconds)
```

```
:: Method 1:
```

```
vol = temporary( vol[*, *, sorted_z] )
```

```
::
```

```
duration1 = SysTime(/Seconds)-t0
```

```
mem1 = (Memory())[3] ; Peak memory usage since last Memory() call
```

```
memUsedMB1 = (mem1-mem0)/1024./1024
```

```
:: Test that it actually worked:
```

```
Print, 'Test 1 '+(Array_Equal(sorted_z, vol[0, 0, *]) ? 'passed' : 'failed')
```

```
vol = IntArr( 512, 512, nPlanes )
```

```
vol[0, 0, *] = IndGen(nPlanes) ; Put [0,1,2...] in position [0,0] on each plane
```

```
mem0 = (Memory())[0] ; Current memory usage
```

```
t0 = SysTime(/Seconds)
```

```
:: Method 2:
```

```
vol2 = IntArr([512, 512, nPlanes], /NoZero)
```

```
FOR planel=0, nPlanes-1 DO vol2[0, 0, planel] = vol[*, *, sorted_z[planel]]
```

```
::
```

```
duration2 = SysTime(/Seconds)-t0
```

```
mem2 = (Memory())[3] ; Peak memory usage since last Memory() call
```

```
memUsedMB2 = (mem2-mem0)/1024./1024
```

```
:: Test that it actually worked:
```

```
Print, 'Test 2 '+(Array_Equal(sorted_z, vol2[0, 0, *]) ? 'passed' : 'failed')
```

Help, duration1, duration2, memUsedMB1, memUsedMB2

END ;; IndexingTest

Output:

```
IDL> indexingtest
% Compiled module: INDEXINGTEST.
Test 1 passed
Test 2 passed
DURATION1    DOUBLE  =    0.89100003
DURATION2    DOUBLE  =    0.67199993
MEMUSEDMB1   FLOAT   =    430.500
MEMUSEDMB2   FLOAT   =    144.000
```

Perhaps you could avoid making the copy of the array if you knew what sequence of swapping planes of your array would lead to the desired sorted_z... left as an exercise for the reader :-)

Cheers,
-Dick

On Tuesday, November 6, 2012 8:05:13 AM UTC-8, rohit bhat wrote:

```
> Hi,
>
> I had a question regarding the use of temporary in IDL.
>
> I wanted to re-order a 3D array.
>
> Suppose the array is vol = IntArr( 512, 512, 287 )
>
> I sorted the Z dimension and got the indices. The command I then used was
>
> vol = temporary( vol[ *, *, sorted_z] )
>
> Despite this, the memory used is still the same if I don't use temporary (almost 5 times the
size of the array)
>
> Is this the correct way to use temporary? If I am using it correctly, is it just the IDL way of
reading columns and rows?
>
> Thanks,
>
> Rohit
```
