Subject: Re: Assignment Time for a 3d Variable
Posted by James Kuyper on Wed, 23 Nov 2005 18:35:21 GMT
View Forum Message <> Reply to Message

David Streutker wrote:
> David Fanning wrote:
>> The speed differences have to do with how you access different
>> parts of the array in memory. If the parts you want are contiguous,
>> then you can get them faster than you can if they are far apart in
>> memory. (Think how much faster it is to pick up the poker
>> chips when they are stacked than when they are scattered all
>> around the table.)
>>
>> To make these kinds of assignments as fast as possible, use
>> the TRANSPOSE function to organize the data into the fastest
>> possible position:
>>
>> IDL> Help, data
>> DATA BYTE = Array[3, 227, 149]
>> IDL> data = Transpose(Temporary(data), [2,3,1])
>> IDL> Help, data
>> DATA BYTE = Array[227, 149, 3]
>
> How does one know which is the fastest possible position? Should the
> largest dimension be first? Nuno's example seems to imply that the
> first dimension is not the fastest accessed.

Note: Nuno's third case was identical to his second one; the time
variation must have been random. With the third case corrected to:

```
for k = 0, 99 do $
      temp = vol[*,*,k]
print, 'execution time (for z axis): '+STRING(systime(1) - time)
```

I get:

```
execution time (for x axis):      0.18765187
execution time (for y axis):     0.038336992
execution time (for z axis):     0.031430006
```

If you're accessing consecutive positions along one dimension of an
object, then the first dimension is in fact the fastest one, because
those consecutive positions are physically adjacent to each other..
That corresponds the fastest case, the third one. The slowest case was
when only one position along the first dimension was accessed for all
possible values along the other two dimensions.