Subject: Re: Dealing with Large data arrays, reducing memory and ASSOC
Posted by Kenneth Bowman on Thu, 14 Jun 2007 18:50:33 GMT
View Forum Message <> Reply to Message

In article <pan.2007.06.14.16.32.00.193280@as.arizona.edu>,
JD Smith <jdsmith@as.arizona.edu> wrote:

> On Thu, 14 Jun 2007 08:08:44 -0500, Kenneth Bowman wrote:
>
>> In article <1181824433.145388.26020@d30g2000prg.googlegroups.com>,
>> Ambrosia_Everlovely <ambrosia_everlovely@hotmail.com> wrote:
>>
>>> [quoted text muted]
>>
>> I would just do it in slices
>>
>> dct = COMPLEXARR(512,512,2048)
>> FOR j = 0, 511 do dct[*,j,*] = FFT(REFORM(dc[*,j,*]), -1, DIM = 2)
>>
>> This does access memory in nearly the worst possible way. If you are
>> going to be doing this a lot, you might want to consider rearranging the
>> data so that t is the first dimension
>>
>> dct = COMPLEXARR(2048,512,512)
>> FOR k = 0, 255 D0 xt[0,0,k] = FFT(REFORM(x[*,*,k]), -1, DIM = 1)
>
> I'd be interested to hear whether this "in order" type of array
> re-arrangement results in a real speedup. I had always assumed this
> is true, but in recent testing on a very different problem, found
> little or no gain, to my surprise.
>
> JD

Here is a quick test that only measures the FFT time:

```
nx =  512
ny =  512
nz = 2048

x  = FINDGEN(nx, ny, nz)
xt = COMPLEXARR(nx, ny, nz)

t = SYSTIME(/SECONDS)

FOR j = 0, ny-1 DO xt[*,j,*] = FFT(REFORM(x[*,j,*]), -1, DIM = 2)

PRINT, 'Time for FFT of 3rd dimension : ', t - SYSTIME(/SECONDS)
```

```
x  = REFORM(x,  nz, nx, ny)
xt = REFORM(xt, nz, nx, ny)

t = SYSTIME(/SECONDS)

FOR k = 0, ny-1 DO xt[0,0,k] = FFT(REFORM(x[*,*,k]), -1, DIM = 1)

PRINT, 'Time for FFT of 1st dimension : ', t - SYSTIME(/SECONDS)
```

I ran it on our new dual quad-core Xeon with 16 GB of memory and got this

```
IDL> @fft_3_test
Time for FFT of 3rd dimension :       -127.68938
Time for FFT of 1st dimension :       -27.563090
```

On my Mac G5 for smaller arrays (256 x 256 x 512) I get this

```
IDL> @fft_3_test
Time for FFT of 3rd dimension :       -4.6950710
Time for FFT of 1st dimension :       -2.5009661
```

I think is a fact of life with cache systems that out-of-order
memory access will cause some penalty.

Ken
