Subject: Re: 2D FFT Slow. Any ideas?
Posted by R.G. Stockwell on Fri, 05 Dec 2003 18:04:40 GMT
View Forum Message <> Reply to Message

"Brian" <brian.huether@NOdlrSPAM.de> wrote in message
news:a298a85e9af4e70d51199dcae50c4c81@news.teranews.com...
> I did a little benchmark between IDL and MATLAB. In each case I created a
> random double precision complex array of size 2048 by 2048 and timed how
> long the 2D FFT took. In MATLAB 6.5 it took about 3.5 sec, and in IDL, it
> took about 10 sec. Is there a way to have IDL use MATLAB for the FFT,
> perhaps using activex? Or would the overhead in using activex defeat the
> purpose?
>
> thanks,
>
> brian
>
>

You could try calling an external routine, some of the best being available
at (fastest ft in the west)

http://fftw.org/

Cheers,
bob

PS I can't believe active x calls would improve speed, but hey, you never
know.
And, I'm surprised that the canned IDL is not very fast. Any chance you
don't really
have a 2048^2 array in idl (did you make a 2049^2 array for instance?)

If you post a blip of example code, I can run them here (matlab and idl) and
verify the time difference,
which would be very interesting.

Subject: Re: 2D FFT Slow. Any ideas?
Posted by K. Bowman on Fri, 05 Dec 2003 18:32:17 GMT
View Forum Message <> Reply to Message

In article <gN3Ab.767$o_.32178@news.uswest.net>,
"R.G. Stockwell" <noemail@please.com> wrote:

> And, I'm surprised that the canned IDL is not very fast. Any chance you
> don't really
> have a 2048^2 array in idl (did you make a 2049^2 array for instance?)

>
> If you post a blip of example code, I can run them here (matlab and idl) and
> verify the time difference,
> which would be very interesting.
>
>

Also, don't forget, IDL only does complex FFTs. Does Matlab do real
FFTs?

Ken Bowman

Subject: Re: 2D FFT Slow. Any ideas?
Posted by R.G. Stockwell on Fri, 05 Dec 2003 21:05:32 GMT
View Forum Message <> Reply to Message

"Brian" <brian.huether@NOdlrSPAM.de> wrote in message
news:a298a85e9af4e70d51199dcae50c4c81@news.teranews.com...
> I did a little benchmark between IDL and MATLAB. In each case I created a
> random double precision complex array of size 2048 by 2048 and timed how
> long the 2D FFT took. In MATLAB 6.5 it took about 3.5 sec, and in IDL, it
> took about 10 sec. Is there a way to have IDL use MATLAB for the FFT,
> perhaps using activex? Or would the overhead in using activex defeat the
> purpose?
>
> thanks,
>
> brian
>

Hi Brian,
I found some time to take a look at this, and I see the same thing you do.
This is on a 1.13 ghz dell inspiron 8100 laptop running win2000.
Matlab 6.5 did the fft of 2048 by 2048 array of doubles in 0.9 seconds.
IDL 6.0 did it in 4.6 seconds (ram 109 MBs).

Wow, that is surprising. The idl version is quite slow.

For a double complex array IDL takes 8.1 seconds (ram 174 MBs),
matlab takes 1.6 sec (211 mb ram).

Interesting.

-bob

Subject: Re: 2D FFT Slow. Any ideas?
Posted by Brian on Mon, 08 Dec 2003 08:02:02 GMT
View Forum Message <> Reply to Message

Hi,

The IDL code is

A=dcomplex(randomn(seed,2048, 2048,/DOUBLE), randomn(seed,2048,
2048,/DOUBLE))
t=systime(1)
FA=fft(A,-1)
print, systime(1)-t

The corresponding matlab code is

A = randn(2048,2048) + i*randn(2048,2048);
tic; FA=fft2(A); toc

I get around 10.4 sec in IDL, and around 3.7 sec in MATLAB 6.5

-brian

"R.G. Stockwell" <noemail@please.com> schrieb im Newsbeitrag
news:gN3Ab.767$o_.32178@news.uswest.net...
> "Brian" <brian.huether@NOdlrSPAM.de> wrote in message
> news:a298a85e9af4e70d51199dcae50c4c81@news.teranews.com...
>> I did a little benchmark between IDL and MATLAB. In each case I created
a
>> random double precision complex array of size 2048 by 2048 and timed how
>> long the 2D FFT took. In MATLAB 6.5 it took about 3.5 sec, and in IDL,
it
>> took about 10 sec. Is there a way to have IDL use MATLAB for the FFT,
>> perhaps using activex? Or would the overhead in using activex defeat the
>> purpose?
>>
>> thanks,
>>
>> brian
>>
>>
>
> You could try calling an external routine, some of the best being
available
> at (fastest ft in the west)
>
> http://fftw.org/
>

> Cheers,
> bob
>
> PS I can't believe active x calls would improve speed, but hey, you never
> know.
> And, I'm surprised that the canned IDL is not very fast. Any chance you
> don't really
> have a 2048^2 array in idl (did you make a 2049^2 array for instance?)
>
> If you post a blip of example code, I can run them here (matlab and idl)
and
> verify the time difference,
> which would be very interesting.
>
>

Subject: Re: 2D FFT Slow. Any ideas?
Posted by Richard French on Mon, 08 Dec 2003 20:17:16 GMT
View Forum Message <> Reply to Message

On 12/8/03 3:02 AM, in article
64b518634e4a83f7473363b5cd15989d@news.teranews.com, "Brian"
<brian.huether@NOdlrSPAM.de> wrote:

> Hi,
>
> The IDL code is
>
> A=dcomplex(randomn(seed,2048, 2048,/DOUBLE), randomn(seed,2048,
> 2048,/DOUBLE))
> t=systime(1)
> FA=fft(A,-1)
> print, systime(1)-t
>
> The corresponding matlab code is
>
> A = randn(2048,2048) + i*randn(2048,2048);
> tic; FA=fft2(A); toc
>
> I get around 10.4 sec in IDL, and around 3.7 sec in MATLAB 6.5
>
> -brian
>

I got FFTW working on a SUNBLADE 1000
IDL> print,!VERSION

{ sparc sunos unix Solaris 6.0 Jun 27 2003 32 64}

(I had to run IDL in 32-bit mode)

For a REAL FFT, 1024x1024, IDL took 2.9 times as long as FFTW
For 2048 x 2048, IDL took 3.4 times as long as FFTW.

On a Dec/HP/Compaq Alpha, the speedup was a factor of 4 for 1024x1024 and about a factor of 5 for 2048x2048.

If you have lots of FFT's to compute, it is worth taking the trouble to build a DLM for the FFTW routines. However, I did not find this easy to do, I used the help from Stein Haugan, and I had to fiddle with the Makefile to get everything working.

I am going to see if I can get fftw to work under MAC OS X, and to build a DLM for that. If anyone has already succeeded in doing this, I'd be interested in knowing how you did it.

Dick French
