
Subject: Re: Different FFT times for same array size ?
Posted by [R.G. Stockwell](#) on Wed, 24 Aug 2005 16:54:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Jan Cami" <jcami@mail.arc.nasa.gov> wrote in message
news:1124862926.625619.103650@g43g2000cwa.googlegroups.com.. .
> Hi,

...
> So the first 28 segments go fast, and from then on things slow down by
> a factor of 50 (well, except those 2 funny ones) !!

Off hand, the factor of 50 increase I would say is due to a
memory "leak" issue. Perhaps at that point, you start to swap to
disk or something. This is just a guess, but often a huge step in
execution time means your process has had to go to the harddrive.
[by leak, it may not be a real leak, but just the fact that
you are running low on memory, and are swapping to disk
to get the rest of the data]

How does it look if you run the same code on the same input
time series (i.e. always fft the same array by
commenting out " input = [input[blen-p+1:blen-1], signal[lower:upper]]").

One thing that i notice is that you are concatenating your new array each
loop.
That may a bit wasteful in memory, although I usually do that with no
problems.

What are the values you use (k, p blen, etc)? Could you include the initial
values that you have just before the loop?
I can run the same code (on a fake signal) and see if I reproduce your
problems.

Cheers,
bob

Subject: Re: Different FFT times for same array size ?
Posted by [jcam](#) on Wed, 24 Aug 2005 18:01:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Bob,

Thanks for the reply. Memory should not be the issue -- I have 2 Gb of
RAM, and it seems like I'm using only about 10% of it.

> Off hand, the factor of 50 increase I would say is due to a

- > memory "leak" issue. Perhaps at that point, you start to swap to
- > disk or something. This is just a guess, but often a huge step in
- > execution time means your process has had to go to the harddrive.
- > [by leak, it may not be a real leak, but just the fact that
- > you are running low on memory, and are swapping to disk
- > to get the rest of the data]

Thanks for the reply. Memory should not be the issue -- I have 2 Gb of RAM, and it seems like I'm using only about 10% of it. The funny part is that for about 90% of the arrays I use, it goes OK, but only in a few cases it goes wrong. Again, array sizes are always the same...

- > How does it look if you run the same code on the same input
- > time series (i.e. always fft the same array by
- > commenting out " input = [input[blen-p+1:blen-1], signal[lower:upper]]").

All short times.... about 0.03 seconds.

- > One thing that i notice is that you are concatenating your new array each
- > loop.
- > That may a bit wasteful in memory, although I usually do that with no
- > problems.

Maybe true, but if the FFT execution time only depends on the size of the arrays, I should get the same type of timing for my different arrays, right ?

- > What are the values you use (k, p blen, etc)? Could you include the initial
- > values that you have just before the loop?

```
IDL> help, k, blen, ip, p, z_filter, input, output
K          LONG      =      151
BLEN       LONG      =     65536
IP         LONG      =     11974
P          LONG      =     53563
Z_FILTER   DCOMPLEX  = Array[65536]
INPUT      DOUBLE    = Array[65536]
OUTPUT     DCOMPLEX  = Array[65536]
```

- > I can run the same code (on a fake signal) and see if I reproduce your
- > problems.

I could also send you two different arrays of the same size, where one causes problems and the other doesn't....

Thanks,

Jan

Subject: Re: Different FFT times for same array size ?

Posted by [jcami](#) on Wed, 24 Aug 2005 18:15:21 GMT

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

A good look at the different input chunks revealed the problem :
there's a few NaN values in there which apparently slow it down
dramatically. Problem solved !

Subject: Re: Different FFT times for same array size ?

Posted by [R.G. Stockwell](#) on Wed, 24 Aug 2005 18:49:21 GMT

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

"Jan Cami" <jcami@mail.arc.nasa.gov> wrote in message
news:1124907321.434895.200100@z14g2000cwz.googlegroups.com.. .

>

> A good look at the different input chunks revealed the problem :
> there's a few NaN values in there which apparently slow it down
> dramatically. Problem solved !

>

interesting! I wasn't aware of that time hit with nans.

Cheers,
bob

Subject: Re: Different FFT times for same array size ?

Posted by [Steve Eddins](#) on Wed, 24 Aug 2005 19:14:18 GMT

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

"R.G. Stockwell" <no@email.please> wrote:

> "Jan Cami" <jcami@mail.arc.nasa.gov> wrote:

>> A good look at the different input chunks revealed the problem :
>> there's a few NaN values in there which apparently slow it down
>> dramatically. Problem solved !

>

> interesting! I wasn't aware of that time hit with nans.

Yes. NaNs tend to slow down floating-point computations substantially. This effect isn't specific to IDL. And with FFTs, every output depends mathematically on every input, so even a single NaN in the input results in every output value being NaN. This is pretty slow way to create an all-NaN

thing, so the original poster might consider treating inputs with NaNs as a special case and handling them differently.

Steve
