

---

Subject: Re: Threads in IDL 7.0

Posted by [Bernhard Reinhardt](#) on Tue, 28 Oct 2008 08:28:42 GMT

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

---

Heinz Stege wrote:

> On Mon, 27 Oct 2008 16:44:29 +0100, Bernhard Reinhardt wrote:

>  
>> Well, I wasn't really precise. I'm not doing this on a 1d-array but on a  
>> 4d-array, where 2 dimensions are time and 2 dimensions are space. I try  
>> to filter special events in time and count those on a 2d-map. Here's the  
>> code:

```
>>  
>>   for i = 0, N_ELEMENTS(STRUC.data[:,0,0,0])-1 do begin  
>>     for j = 0, N_ELEMENTS(STRUC.data[0,:,0,0])-1 do begin  
>>       indices = Where(STRUC.data[i,j,*,*] GE 150., count)  
>>       freq [i,j]=count  
>>     endfor  
>>   endfor
```

```
>>  
>> Although the array data is quite big "where" only gets a small portion  
>> to see of it. So thread-pool isn't invoked. => CPU-Usage still 50%
```

```
>>  
>> I also asked some more IDL-experienced colleagues about generating  
>> threads manually but they also didn't know about anything like that :(
```

```
>>  
>> BUT using your method still brought me a gain of 3.6 times faster  
>> execution :)
```

```
>>  
>> regards
```

```
>>  
>> Bernhard
```

>  
> Please try the following command:

```
>  
>   freq=total(total(STRUC.data ge 150.,4,/integer),3,/integer)
```

```
>  
> The IDL manual says, that TOTAL makes use of IDL's thread pool. And  
> there is no for-loop needed anymore...
```

Now that's fast! Who cares about threads? ;) Thanks to everyone.

Regards

Bernhard

---