Subject: Re: Was: Index... Now: Vectorize, huh?
Posted by Pavel A. Romashkin on Thu, 05 Apr 2001 21:13:06 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

> Part of the
> problem is that you coded your vectorized path inefficiently.

I tried it the way you did, but decided to try the full "one-line"
vectorized solution.

> I've always said that if you can vectorize the inner loop of your
> operation then you are usually fine. Pavel, you actually did that in
> your *non*-vectorized case. :-)

This is true, and I do this in my code since I found that out a while
ago. I wonder if that means, in fact, that array operations in IDL are
optimized for vectors (row-wise arrays) only, and once you are into more
than 1 dimension, you are better off looping through other dimensions.

> A new version of TEST improves things slightly, but doesn't tip the
> scales. w/ your version I get 1.6 and 3.8 s. With my version I get
> 1.5 and 2.7 s.

On my system, your own solution from the previous post is faster still,
but the loop can not be defeated:

```
;***********
function test, s, vec=vec
start = systime(1)
x = findgen(s)
if keyword_set(vec) then begin
;a = rebin(x, s, s)^2
;a = sqrt(transpose(a) + a)
;a = sqrt((transpose(rebin(x, s, s)))^2 + (rebin(x, s, s))^2)
a = (x # (fltarr(s)+1))^2
a = sqrt(transpose(a) + a)
endif else begin
a = fltarr(s, s)
for i = 0, s-1 do a[0, i] = sqrt(x^2 + i^2.)
endelse
print, systime(1) - start
return, a
end
;***********
```

Cheers,
Pavel