Subject: Re: vectorization challenge! (help!)
Posted by alvin[1] on Thu, 19 Jul 2007 08:01:55 GMT
View Forum Message <> Reply to Message

On Jul 17, 9:52 pm, Conor <cmanc...@gmail.com> wrote:
> I'm 'vectorizing' a piece of code to speed it up. It's part of a
> larger program. One of the sections is turning out to be very
> difficult to vectorize (am I using that word right? What I mean is
> I'm trying to get rid of for loops). Anyway, maybe someone has some
> thoughts on how to vectorize it. Maybe it's just not worth it for
> this section. Here's the basic idea, filled in with dummy data:
>
> n = 24
> npeeps = 50
> gn1 = findgen(n,npeeps)
> gn2 = findgen(n,npeeps)
> cutoff = .01
>
> for i=0,npeeps-1 do begin
>
> ; make a random value to determine if we do anything with this
> row
> if randomu(seed,1) lt cutoff then begin
>
> ; this row has been selected. Swap the last (random number)
> of digits in gn1[*,i] with gn2[*,i]
> randindex = long(randomu(seed,1)*n*nd)
> temp = gn1[randindex:*,i]
> gn1[randindex:*,i] = gn2[randindex:*,i]
> gn2[randindex:*,i] = temp
>
> endif
>
> endfor
>
> That's it. For randomly selected rows, swap a random number of
> elements at the end of the row with another array. It is surpisingly
> difficult to get rid of that for loop. Maybe I'm just a bit out of it
> today though. I thought of generating a list of indexes to be
> swapped, but I can't quite figure it out. Oh, if only IDL allowed the
> syntax: arr[st:ed] where st and ed are arrays themselves! Then this
> would be really easy (something like this would do it):
>
> st = long(randomu(seed,npeeps)*n*nd)
> ed = make_array(npeeps,/integer,value=n)
> indlist = indgen(n)
> inds = indlist[st:ed] + indgen(npeeps)*n
> temp = gn1[inds]

> gn1[inds] = gn2[inds]
> gn2[inds] = temp
>
> Alas, indlist[st:ed] isn't allowed! (Also, indgen(npeeps)*n has the
> wrong dimensions anyway...)

Hi there:

If I understand your problem truly, I would have done this:
The code is not efficient, and contains redundant values in the 'for'
loop.
With a little playing around, you can reduce the size of randindex in
the loop.

Alvin

```
n=24         ; How large is this value?
npeeps=50    ;How large is this?
gn1 = findgen(n,npeeps)
gn2=qn1

;;;;;cutoff = .01 ;;;; What does this do?

randindex = long(randomu(seed,npeeps)*n)          ;;;;; I don't know
what 'nd' is!

;;;if randomu(seed,1) lt cutoff then begin  ??    ;;;;This is throwing
me off...
                                          ;;;;What do you
intend to do here?

thisval=n*indgen(npeeps)+n-1
for i=1,n do randindex=[randindex,(randindex+n-1)<thisval]
temp = gn1[randindex]
gn1[randindex] = gn2[randindex]
gn2[randindex] = temp
```