Subject: comparing and concatenating arrays...please help!!

Posted by m.doyle on Thu, 08 Jan 2004 10:27:57 GMT

Hello all,

I really hope someone out there can help me with this....I am tearing my hair out as my code is so slow!

I have 2 files of data (hourly met data) with one file containing one set of parameters, and the other file containing another set of parameters. What I am trying to do, is to match the data based on the YY, MM, DD and HH values and then write BOTH sets of parameters to a seperate file. For example;

file1:
1954 12 31 23   90   11   4 366   0.00

file2:
1954 12 31 23   2.80   2.10   2.20   95.21

intended result:
1954 12 31 23   90   11   4 366   0.00 2.80   2.10   2.20
95.21

NOTE: Both files have no order to them, so a simple concatenation won't work

I have written some code, but it is wrist slashing-ly slow!;

I read in each variable as a seperate array...

```
b=0L
REPEAT BEGIN
c=0L
REPEAT BEGIN
If (year(b) EQ year2(c)) AND (month(b) EQ month2(c)) AND (day(b) EQ
day2(c)) AND (hour(b) EQ hour2(c)) THEN BEGIN

printf, 3, year(b), month(b), day(b), hour(b), winddir(b), windsp(b),$
 present(b),visib(b), mslpres(b), airt(c), dewt(c), wett(c), relh(c),$
 format = finalformat
endif

c=c+1

ENDREP UNTIL c EQ lines2-1
```

b=b+1

ENDREP UNTIL b EQ lines1-1

I'm sure there must be a better way than this.

Please help me!

Many thanks in advance, Martin..

Subject: Re: comparing and concatenating arrays...please help!!
Posted by JD Smith on Fri, 09 Jan 2004 18:12:20 GMT
View Forum Message <> Reply to Message

On Thu, 08 Jan 2004 03:27:57 -0700, Martin Doyle wrote:

> Hello all,
>
> I really hope someone out there can help me with this....I am tearing my
> hair out as my code is so slow!
>
> I have 2 files of data (hourly met data) with one file containing one
> set of parameters, and the other file containing another set of
> parameters. What I am trying to do, is to match the data based on the
> YY, MM, DD and HH values and then write BOTH sets of parameters to a
> seperate file. For example;
>
> file1:
> 1954 12 31 23 90 11   4 366   0.00
>
> file2:
> 1954 12 31 23   2.80   2.10   2.20   95.21
>
> intended result:
> 1954 12 31 23 90 11   4 366   0.00 2.80   2.10   2.20 95.21
>
> NOTE: Both files have no order to them, so a simple concatenation won't
> work
>
> I'm sure there must be a better way than this.

I predict this can be done in IDL in under 3 seconds. This is easy to
convert into an "intersection of two arrays" problem: as Ben suggests,
convert year, month, day, hour into a single long integer number
(could be julian hours, could be hours since Jan 1, 1970, a long with
all the data encoded in different bits, whatever). Read the entire

file in at once (READCOL comes to mind) into separate vectors for each
column, and perform this date conversion on the first 4. You now have
two long integer vectors you'd like to match up, call them A and B.
Read up on the various list intersection methods:

 http://groups.google.com/groups?selm=38CBF8B6.5BF0AB50%40ast ro.cornell.edu

The last paragraph gives a nice synopsis of which to use: I'd expect
either the SORT or HISTOGRAM methods will work. Stay away from the
ARRAY method for such large data sizes. Your problem has one
additional wrinkle: you don't just want the indices in A which exist
anywhere in B, you also want the matching indices in B. The HISTOGRAM
method seems ideally suited to this, especially if your data come in
regularly every hour, i.e. are not sparse (sometimes with an interval
of an hour, sometimes two weeks), with a simple modification to
capture the B indices:

```
function ind_int_HISTOGRAM, a, b, WHERE_B=whb
  minab = min(a, MAX=maxa) > min(b, MAX=maxb)
  maxab = maxa < maxb
  ha = histogram(a, MIN=minab, MAX=maxab, REVERSE_INDICES=reva)
  hb = histogram(b, MIN=minab, MAX=maxab, REVERSE_INDICES=revb)
  r = where((ha ne 0) and (hb ne 0), cnt)
  if cnt eq 0 then return, -1
  if arg_present(whb) then whb=revb[revb[r]]
  return,reva[reva[r]]
end
```

I tried this on two 250,000 long integer vectors which were about 1 in
4 sparse, and it took less that 1/2 second on my feeble laptop, which
should nicely beat a Perl hash for data this regular (sparser or more
random data is another story -- hashes are ideally suited for that):

```
IDL> b=long(randomu(sd,250000L) * 1000000L)
IDL> a=long(randomu(sd,250000L) * 1000000L)
IDL> t=systime(1) & wha=ind_int_HISTOGRAM(a,b,WHERE_B=whb) & print,systime(1)-t
      0.42473805
```

Also, if you want all the indices which are not in both A and B, look
into the COMPLEMENT keyword to where, and use it in both instances
above to return a WHERE_ONLY_A and WHERE_ONLY_B keyword in the same
fashion.

JD

Subject: My final solution..thanks for your help!

Hello everyone, and many, many thanks for all your helpful suggestions.

I managed to get the runtime for this problem down to 35 seconds, from what I previously estimated was going to take about 4 days using for and if loops! Good old IDL!

I ended up combining many of the solutions posted previously, as follows;

My original files were in the format below:

> file1:
> 1954 12 31 23 90 11   4 366   0.00
>
> file2:
> 1954 12 31 23   2.80   2.10   2.20   95.21

With the intended result:
> 1954 12 31 23 90 11   4 366   0.00 2.80   2.10   2.20   95.21

I used Ben's suggestion and concatenated the first 4 columns of each file resulting in a "field ID" if you like:

> file1_ID= (file1[0,*]*1000000D) + (file1[1,*]*10000D) +
(file1[2,*]*100D) + >file1[3,*]
> file1_ID_final = round([file1_ID])

result: 1954123123

I then used the match() routine from the NASA library:

 http://groups.google.co.uk/groups?selm=331C553A.41C67EA6%40a
strosun.tn.cornell.edu&oe=UTF-8&output=gplain

This program allowed me to output 2 vectors of indices indicating matching pairs of "field ID's". These outputs were suba for file1 and subb for file2. For example, if suba[0] = 2 and subb[0] = 5, then file1_ID[2] EQ file2_ID[5].

I then concatenated the 2 files based on these indices;

> endresult = [file1(*,suba(*)), file2(4,subb(*)),file2(5,subb(*)),
file2(6,subb(*)), file2(7,subb(*))]

and output!

> printf, 3, endresult, format = finalformat

Once again, many thanks for all your helpful suggestions,

Best wishes,

Martin..

m.doyle@uea.ac.uk (Martin Doyle) wrote in message
news:<d33d6a4b.0401080227.1a588e88@posting.google.com>...
> Hello all,
>
> I really hope someone out there can help me with this....I am tearing
> my hair out as my code is so slow!
>
> I have 2 files of data (hourly met data) with one file containing one
> set of parameters, and the other file containing another set of
> parameters. What I am trying to do, is to match the data based on the
> YY, MM, DD and HH values and then write BOTH sets of parameters to a
> seperate file. For example;
>
> file1:
> 1954 12 31 23 90 11   4 366   0.00
>
> file2:
> 1954 12 31 23   2.80   2.10   2.20   95.21
>
> intended result:
> 1954 12 31 23 90 11   4 366   0.00 2.80   2.10   2.20
> 95.21
>
> NOTE: Both files have no order to them, so a simple concatenation
> won't work
>
> I have written some code, but it is wrist slashing-ly slow!;
>
> I read in each variable as a seperate array...
>
> b=0L
> REPEAT BEGIN
> c=0L
> REPEAT BEGIN
> If (year(b) EQ year2(c)) AND (month(b) EQ month2(c)) AND (day(b) EQ
> day2(c)) AND (hour(b) EQ hour2(c)) THEN BEGIN
>

```
> printf, 3, year(b), month(b), day(b), hour(b), winddir(b), windsp(b),$
>  present(b),visib(b), mslpres(b), airt(c), dewt(c), wett(c), relh(c),$
>  format = finalformat
> endif
>
> c=c+1
>
> ENDREP UNTIL c EQ lines2-1
>
> b=b+1
>
> ENDREP UNTIL b EQ lines1-1
>
> I'm sure there must be a better way than this.
>
> Please help me!
>
> Many thanks in advance, Martin..
```