Subject: Re: comparing and concatenating arrays...please help!!
Posted by Pepijn Kenter on Thu, 08 Jan 2004 13:25:13 GMT
View Forum Message <> Reply to Message

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 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..

Hi.

You'll need a more efficient algorithm. For each line in file1 you walk through all the data of file2. This costs in the order of lines1*lines2 operations (btw, how big are these files?). This means that if these files double in size, your program will run 4 times as long!

I'm sure that your program can be speeded up with some smart use of the WHERE command, but since the WHERE command also traverses through a complete array, nothing is changed in principle.

To do better than that you first have to sort the data. You can use the SORT procedure of IDL. I don't know what algorithm IDL uses, but in general sorting a dataset with n elements can be done in the order of n*log(n) operations (instead of n^2, what you use now). Furthermore, a lot of effort is put in this routine to make it as efficient as posible; let IDL do the hard work. You could also use an external program to sort your files, like the sort command under linux.

When you have sorted the data, you'll need to write an algorithm that traverses both arrays simultaniously. For example, walk through dataset1 and for each line in set1 search the line in the set2 with the same date starting at the previous found line in set2. Because your files are sorted, you only need to walk trough a small part of file2 for each line in file1. I'm sure you can think of something.

HTH, Pepijn Kenter.

PS. please indent your code, this makes it more readable.

Subject: Re: comparing and concatenating arrays...please help!!
Posted by m.doyle on Thu, 08 Jan 2004 20:26:12 GMT
View Forum Message <> Reply to Message

Hi Pepijn,

Many thanks for your reply.

My files are _huge_. File 1 is about 250000 lines and file2 about half that. which is why my code was taking about 4 days!

I'll give the sorting a try, but if anyone else has any suggestions, they'll be gratefully received. I'm not a newbie, but am daunted by some of the operations you suggested below... I certainly take inspiration from all you guys though!

Al the best, Martin

Pepijn Kenter <kenter_remove_spam@tpd.tno.nl> wrote in message news:<3FFD5A39.9000603@tpd.tno.nl>...
> 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 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..
>
> Hi.
>
> You'll need a more efficient algorithm. For each line in file1 you walk
> through all the data of file2. This costs in the order of lines1*lines2
> operations (btw, how big are these files?). This means that if these
> files double in size, your program will run 4 times as long!
>
> I'm sure that your program can be speeded up with some smart use of the
> WHERE command, but since the WHERE command also traverses through a
> complete array, nothing is changed in principle.
>
> To do better than that you first have to sort the data. You can use the
> SORT procedure of IDL. I don't know what algorithm IDL uses, but in
> general sorting a dataset with n elements can be done in the order of
> n*log(n) operations (instead of n^2, what you use now). Furthermore, a
> lot of effort is put in this routine to make it as efficient as posible;
> let IDL do the hard work. You could also use an external program to
> sort your files, like the sort command under linux.
>
> When you have sorted the data, you'll need to write an algorithm that
> traverses both arrays simultaniously. For example, walk through dataset1
> and for each line in set1 search the line in the set2 with the same date
> starting at the previous found line in set2. Because your files are
> sorted, you only need to walk trough a small part of file2 for each line
> in file1. I'm sure you can think of something.
>
> HTH, Pepijn Kenter.

>
> PS. please indent your code, this makes it more readable.

Subject: Re: comparing and concatenating arrays...please help!!
Posted by Michael Wallace on Thu, 08 Jan 2004 21:22:40 GMT
View Forum Message <> Reply to Message

> Hi Pepijn,
>
> Many thanks for your reply.
>
> My files are _huge_. File 1 is about 250000 lines and file2 about half
> that. which is why my code was taking about 4 days!
>
> I'll give the sorting a try, but if anyone else has any suggestions,
> they'll be gratefully received. I'm not a newbie, but am daunted by
> some of the operations you suggested below... I certainly take
> inspiration from all you guys though!

I'd try the sorting attack myself -- sort first and then merge the sorted data. I don't know what kind of memory and processing constraints you're under, so this might not be applicable if you have sufficient memory. If you run into a memory problem trying to sort this much data, you can divide the sort into chunks. Read and sort the first X many lines, then read and sort the next X many, and so on. Then take all the chunks and merge them. Finally, merge of the sorted file1 and sorted file2. While there will be more operations than the first approach, the running time will still be on the order of n*log(n) which is still better than your original running time of n^2, and you will chew up a lot less memory. Again, this is just a suggestion if you run into memory problems trying to do such a large sort.

Mike W

Subject: Re: comparing and concatenating arrays...please help!!
Posted by Pepijn Kenter on Thu, 08 Jan 2004 22:57:03 GMT
View Forum Message <> Reply to Message

>
> My files are _huge_. File 1 is about 250000 lines and file2 about half
> that. which is why my code was taking about 4 days!

You also might want to consider using another programming language. IDL is designed for doing (scientific) calculations, not for a task like this. Loop constructs are particularly slow in IDL.

I can imagine that if you're in a hurry, you start programming in whatever language you're most familiar with (I'm also lazy in that respect ;-). But if you've got the time, a project like this is a good opportunity to learn a new language. IMHO, the only reason for using IDL for a task like this would be when the input is generated by another IDL program (or the output is used by an IDL program). In that case the programs could be integrated thightly if necessary.

Languages designed for processing text files are e.g. perl and awk. If speed is really important you can use C for everything; but I guess it would take a lot of effort to beat a specialized language, so better try these first.

And like I said before, the sort program is very fast in sorting text files. I've just generated a 480000 line test file. It took 11 seconds to sort on an atlon 1200.

>
> I'll give the sorting a try, but if anyone else has any suggestions,
> they'll be gratefully received. I'm not a newbie, but am daunted by
> some of the operations you suggested below... I certainly take
> inspiration from all you guys though!
>

If you want to read some more about computational complexity, check out:

http://en2.wikipedia.org/wiki/Computational_complexity_theor y
http://users.forthnet.gr/ath/kimon/CC/CCC1b.htm

Pepijn.

Subject: Re: comparing and concatenating arrays...please help!!
Posted by Michael Wallace on Thu, 08 Jan 2004 23:36:58 GMT
View Forum Message <> Reply to Message

> Languages designed for processing text files are e.g. perl and awk. If
> speed is really important you can use C for everything; but I guess it
> would take a lot of effort to beat a specialized language, so better try
> these first.

Perl is a good language for everyone to know, especially with regards to text processing. There have been plenty of times I've used Perl to reorder, massage, redistribute and otherwise mangle data files before IDL ever sees them.

Sometimes it does get a bit obfuscated if you let it. There was one

time I had a bash shell script which called some perl programs which
then called some Java programs which then interacted with a database and
wrote some files which then got immediately processed by IDL. The end
results were just some plain text data files and images, but it was a
circuitous path at best. But, it was a fun programming adventure!! ;-)

Mike

Subject: Re: comparing and concatenating arrays...please help!!
Posted by wmconnolley on Fri, 09 Jan 2004 09:50:37 GMT
View Forum Message <> Reply to Message

Martin Doyle <m.doyle@uea.ac.uk> wrote:
> My files are _huge_. File 1 is about 250000 lines and file2 about half
> that. which is why my code was taking about 4 days!

Here is my initial entry in the obfusticated solution using perl.
However, I think I misread you because in whats below I've assumed
that dates for each ob exist in each file. If you only want output
when both match, and throw away what doesn't have equiv's in both files,
then use the alternative last line.

```
#!/bin/perl

@f1=split("\n",`cat f1`);
@f2=split("\n",`cat f2`);

for (@f1) { ($y,$m,$d,$h,$r)=(/(....) (..) (..) (..) (.*)/); $f1{"$y $m $d $h"}=$r };
for (@f2) { ($y,$m,$d,$h,$r)=(/(....) (..) (..) (..) (.*)/); $f2{"$y $m $d $h"}=$r };

for $k (sort keys %f1) { print "$k $f1{$k} $f2{$k}\n" };

# Alternative output line
# for $k (sort keys %f1) { if ($f1{$k} and $f2{$k} ) { print "$k $f1{$k} $f2{$k}\n" } };
```

This should be fast (a few mins at most) if all the data fits into your memory at
once, as it ought to. You can write "reverse" in front of sort if you want it
backwards.

-W.

--
William M Connolley | wmc@bas.ac.uk | http://www.antarctica.ac.uk/met/wmc/
Climate Modeller, British Antarctic Survey | Disclaimer: I speak for myself
I'm a .signature virus! copy me into your .signature file & help me spread!

Subject: Re: comparing and concatenating arrays...please help!!
Posted by btt on Fri, 09 Jan 2004 15:30:23 GMT
View Forum Message <> Reply to Message

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 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..

Hello,

You have quite a few non-IDL options presented already. But if you're stuck in IDL as I am then I have a suggestion... at least a tempting place to start.

I would collapse the date/time information into a julian day number. Then I would histogram the julian day numbers which is the same as sorting.

```
 ;merge the year/month/day data of the files
month = MonthsFile1 + MonthsFile2
day = DayFile1 + DayFile2
year = YearFile1 + YearFile2

 ;convert to long integer Julian day number
jul = JULDAY(month, day, year) ; note leave the hour out for now!

 ;compute histogram - resereve the reverse indices
H = Histogram(jul, reverse_indices = r)
```

Now each occupied bin contains the data for a single day (all 24 hours) and I'd bet they are listed in chronological order, too. You'll have to sort through the hours - but you can keep track of the two different data sources since any reverse_index GT n_elements(file1)-1 must have come from file2.

Honestly, I don't know if this would be a savings for you or not since I would expect a pretty flat histogram. JD Smith has written up a tutorial on this very subject and it can be found at the fountain of youth and tennis (www.dfanning.com).

Come to think of it, you might be able to modify the Julian Day number to include the hours.... jul = (jul * 100) + hours. But then again, it might squander the efficiency gain using Histogram.

Cheers,
Ben

Subject: Re: comparing and concatenating arrays...please help!!
Posted by R.Bauer on Fri, 09 Jan 2004 17:55:10 GMT
View Forum Message <> Reply to Message

Martin Doyle wrote:

Dear Martin,

we have some routines which could be used to synchronize your data.

With string2js you could convert your data columns of time into julian
seconds.

for example:
time1=string2js('1954 12 31 23',format=' Y M D H')

If you have done this you have a numerical time

print,js2string(-1.4200740d+09)
gives 1954-12-31 23:00:00 000

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/string2js_dbase.pro.html
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/js2string_dbase.pro.html

You may be interested in our time_series_sync routine too

http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl
_html/dbase/time_series_sync_dbase.pro.html

Result = TIME_SERIES_SYNC(Master_time, Client_time, Client_value)

If you are able to store your data in an icg-data-structure you could do
this for all parameters at once by icg_ts_sync

if you are interested this may be interesing
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/gen_icgs_dbase.pro.html

s=gen_icgs(/small,short=['time','P1','P2'])
help,s,/str
*s.time.param=string2js(time,format=' Y M D H')
*s.time.units='seconds since 2000-01-01 00:00:00'
*s.time.long_name='time'
*s.p1.param=
*s.p1.units=
*s.p1.long_name=
*s.p2.param=

*s.p2.units=
*s.p2.long_name=
s=chk_struct(s)
master=ptr_struct2struct(s,/free)

result=icg_ts_sync(master,client)
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_source/idl _html/dbase/icg_ts_sync_dbase.pro.html

This structure could be stored to several data formats already.
e.g. netCDF, HDF, nasa FFI 1001, END

For further routines and licensing please have a look at
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html

best regards

Reimar

> 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..

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-I)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
---------- -------
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html
========== =======