Subject: Re: Where vs Histogram vs ??
Posted by R.G. Stockwell on Thu, 17 Oct 2002 13:18:35 GMT
View Forum Message <> Reply to Message

Andrew Cool wrote:
> Hello All,
> At the moment I'm doing something like this :-
>
> start_year = 2000
> end_year = 2002
> start_day = 120
> end_day = 133
> start_half_hr = 0
> end_half_hr = 47
> WRF = 1
> FREQ = 2
> start_beam = 0
> end_beam = 3
> nominated_parameter = 2
>
> index = Where(!database.year GE start_year AND $
> !database.year LE end_year AND $
> !database.day GE start_day AND $
> !database.day LE end_day AND $
> !database.beam GE start_beam AND $
> !database.beam LE end_beam AND $
> !database.half_hr GE start_half_hr AND $
> !database.half_hr LE end_half_hr AND $
> !database.WRF EQ WRF AND $
> !database.FREQ EQ FREQ AND $
> !database.parameter(nominated_parameter) NE
> bad_data_value)
>

A quick suggestion, change year, day, and half_hr into
julian day number, and thus reduce three searches into one.

index = Where((!database.time GE start_time) and (!database.time LT end_time),count)

BUt, if you REALLY want eye crossing speed, don't search on the array of structures.
Create an array of julian days (of the same size as database) and search that
for the index with which to use on the database array.
Here are the time differences, the code snippet that produced it is below.

IDL> .GO
time elapsed for where function array of struct 0.14732301
time elapsed for where function array of struct 0.036754966

DATABASE   STRUCT   = -> <Anonymous> Array[999999]

Another suggestion, you could always make your data base
a "real" database and use SQL style queries. This would
require some programming outside IDL (or getting Dataminer, etc.)

Cheers,
bob stockwell

PS
A small note, you use "GE start" and "LE end", so you may possibly
include the "end" in two subsets of the data (i.e. as the highest
in one subset,and as the lowest in the next subset). You might
want to look at using "GE start" and "LT end" (note LT rather than
LE). This would only be a problem if you are counting through
starttimes in a loop or something like that.

```
; START CODE SNIPPET THAT COMPARES A WHERE OF an array of
;  structures, vs a simple array


data_st = {YEAR      : 0          ,$
                DAY       : 0          ,$ ; 136 days over 12 years
                HALF_HR   : 0          ,$ ; 0..47
                RANGE_IDX : 0          ,$ ; 0..267
                WRF       : 0B         ,$ ; 3 possible values
                FREQ      : 0B         ,$ ; 4 possible values
                BEAM      : 0B         ,$ ; 4 possible values
                PAD       : 0B         ,$ ; Padding to align byte
                Parameter : FLTARR(5)}
len = 999999L

database = Replicate(data_st,len)
array = dblarr(len)

for i = 0L,len-1 do begin
 database[i].year = (randomn(seed,1))[0]
 array[i] =  (randomn(seed,1))[0]
endfor
```

```
t0 = systime(1)
w = where(database.year gt 0.5,count)
print,'time elapsed for where function array of struct',-t0 + systime(1)

t0 = systime(1)
w = where(array gt 0.5,count)
print,'time elapsed for where function array of struct',-t0 + systime(1)
```

Subject: Re: Where vs Histogram vs ??
Posted by David Fanning on Thu, 17 Oct 2002 13:58:18 GMT
View Forum Message <> Reply to Message

Andrew Cool (andrew.cool@dsto.defence.gov.au) writes:

```
>    I have a structure defined as :-
>
>   data_st = {YEAR      : 0         ,$
>                  DAY       : 0         ,$ ; 136 days over 12 years
>                  HALF_HR   : 0          ,$ ; 0..47
>                  RANGE_IDX : 0          ,$ ; 0..267
>                  WRF       : 0B         ,$ ; 3 possible values
>                  FREQ      : 0B         ,$ ; 4 possible values
>                  BEAM      : 0B         ,$ ; 4 possible values
>                  PAD       : 0B         ,$ ; Padding to align byte
>  boundaries
>                  Parameter : FLTARR(5)}
>
>
>    Replicate that a few times :-
>
>   database = Replicate(data_st,15425228)
```

15425228!? The mind boggles. :-(

In a *structure*!? I guess it would be slow. I think I would
lean toward some kind of flat file structure that could be accessed
quickly with associated variables and an array (arrays?) of
pointers.

```
>      Is there a quicker way than the above monsterous Where statement?
>      I've browsed the Histogram tut on David Fanning's site, and rapidly found
>      my eyes glazing over. Can Histogram help here? Perhaps multiple nested
>      Histograms? David's SetUnion or SetIntersection, maybe?
```

SetUnion and SetIntersection are just fancy wrappers to Histogram.

Cheers,

David

P.S. Let's just say I'm pretty sure my lack of good ideas
this morning has to do with the excessive celebration after
beating that 25 year-old kid again last night. (On a service
ace and an overhead, if you can imagine!) Oooohh, that kid
hates to play me. But I think that is the last time he will
call me "Grandpa." :-)

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Where vs Histogram vs ??
Posted by Andrew Cool on Thu, 17 Oct 2002 23:29:20 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Andrew Cool (andrew.cool@dsto.defence.gov.au) writes:
>
>> I have a structure defined as :-
>>
>> data_st = {YEAR : 0 ,$
>> DAY : 0 ,$; 136 days over 12 years
>> HALF_HR : 0 ,$; 0..47
>> RANGE_IDX : 0 ,$; 0..267
>> WRF : 0B ,$; 3 possible values
>> FREQ : 0B ,$; 4 possible values
>> BEAM : 0B ,$; 4 possible values
>> PAD : 0B ,$; Padding to align byte
>> boundaries
>> Parameter : FLTARR(5)}
>>
>>
>> Replicate that a few times :-
>>
>> database = Replicate(data_st,15425228)
>
> 15425228!? The mind boggles. :-(
>
> In a *structure*!? I guess it would be slow. I think I would
> lean toward some kind of flat file structure that could be accessed

> quickly with associated variables and an array (arrays?) of
> pointers.

Hi David,

As always, there's some history behind these things.

The original data was held in 1536 files, where the filenames
were derived from year/day/freq/wrf, and each file held data
by beam and range.

Thus to generate a plot of some selection of parameters for the
entire database required sequentially opening and reading 1536 files,
and building up histograms or whatever from each file.
(Since we last met, the gray hair I've collected is partially due to
to waiting for these files to be read in...)

At 2GB over 1536 files, the entire database was too big to be held
in memory at the one time. The guy who developed the database in
'94 also included all the "blanks" for which there was no data
collected.

By excluding the blanks, and using the smallest possible datatype for
each variable, I've condensed the database to 470MB - Now we can have
the whole thing in memory after a 3-4 minute load!!

What I did neglect to say in my first post was that as I created the
condensed database, I also created an index of where each of the 136
days starts/ends. So my monsterous Where constructions are acting on
about 15,425,228 / 136 (113421) records per day on average. Even so,
it still takes 10+ minutes.

The collection of data looks a bit like this :-

day 1..136
  half hour 0..47 max, typically about 24 half hour blocks used
          Beam normally cycled 0,1,2,3,0,1,2,3...
            WRF normally cycled 0,1,2,0,1,2...
              FREQ normally cycled 0,1,2,3,0,1,2,3...
                Range max 267 cells, typically 120-150
                  Data values 5 for every rng/freq/wrf/beam/hh/day

There's no such thing as a fixed record length, as the number of
half hours varies from day to day, the number of ranges varies
dependent upon freq and WRF, and the freq, WRF and beam may have been
manually set on the day of data collection rather than normally cycled.

Hence I opted not to use a flat file - I think it would be quite hard
to index into a flat file given so many variable lengths.

Whereas a structure handles most of the indexing for you. Its just
fairly #$%^# slow when you're talking 15,425,228 records...

Maybe now that I've got the cake in memory, wanting to eat it too
entails getting sticky fingers?

Andrew

> P.S. Let's just say I'm pretty sure my lack of good ideas
> this morning has to do with the excessive celebration after
> beating that 25 year-old kid again last night. (On a service
> ace and an overhead, if you can imagine!) Oooohh, that kid
> hates to play me. But I think that is the last time he will
> call me "Grandpa." :-)

Just what brand of hair tonic/dye are you using??

> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

--

---------- ----------
Andrew D. Cool .->-.
Electromagnetics & Propagation Group '-<-'
Intelligence, Surveillance & Reconnaissance Division Transmitted on
Defence Science & Technology Organisation 100% recycled
PO Box 1500, Edinburgh electrons
South Australia 5111

Phone : 061 8 8259 5740 Fax : 061 8 8259 6673
Email : andrew.cool@no-spam.dsto.defence.gov.au
Please remove the no-spam from my email address to reply
---------- ----------