Subject: point_lun is slow
Posted by George McCabe on Wed, 27 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

hello,

reading from a data file at regularly spaced byte locations, 2 bytes at
a time using point_lun - my program is abnormally slow.  i don't have
enough experience to guess whether the poor performance is inherent to
point_lun & readu approach or if there are options which are affecting
execution adversely.

i'ld appreciate any thoughts on the topic which might lead to a
solution.

thank you,

george mccabe, gmccabe@replica.gsfc.nasa.gov

Subject: Re: point_lun is slow
Posted by Liam Gumley on Thu, 28 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

George McCabe wrote:
> chunking is certainly much faster, and my algorythm is 'chunking' away
> nicely.
>
> in instances where a small number of the total data elements from the
> file are required, the 'chicken pecking' approach is much faster.  but
> when in doubt, chunk.

I believe you can use the 'chunking' method in both cases with high
efficiency. The key here is to access the disk in sequential order, with
as few disk accesses as possible. I'm assuming that the goal is to read
small (say 2 byte) sections of data from the disk at random locations.

The following pseudo-algorithm reads records (chunks) of data from the
disk in sequential order. Only records that cover the specified read
locations are actually read from disk. Each record is only read once.

Sort the array of read locations from lowest to highest
Set the record size to 512 bytes (you can experiment with record sizes)
Set the old record number to -1
Start a loop over the read locations
  For this read location, compute the record number in the file
  If the record number is different than the old record number
    Read the current record

Set the old record number to the current record number
End If
For this read location, compute the byte offset within the record
Extract data from the record at the byte offset
End Loop

This method should be just as efficient for small or large numbers of
read locations.

Cheers,
Liam.

--
Liam E. Gumley
Space Science and Engineering Center, UW-Madison
http://cimss.ssec.wisc.edu/~gumley

---

Subject: Re: point_lun is slow
Posted by George McCabe on Thu, 28 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Struan Gray wrote:

> simplest way to access this data be to use ASSOC to associate a huge
> array of 2-byte variables with the file and then subscript it as
> necessary to read in either frames or groups of individual pixels.

Struan,

ASSOC - certainly the simplest and also the fastest in cases where every
element of the data is needed. but when for example only the
[1538,395778,790018,... 434454018] location elements are needed from the
file it is faster to point_lun to each. furthermore, the size of each
associated read can affect the overall speed of a procedure. it's a
better understanding of this subtlety that i am hoping for.

george

---

Subject: Re: point_lun is slow
Posted by Struan Gray on Thu, 28 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

George McCabe, george.mccabe@gsfc.nasa.gov writes:

> reading from a data file at regularly spaced byte
> locations, 2 bytes at a time using point_lun
> - my program is abnormally slow.

Perhaps I'm just being dim (it has been known :-), but wouldn't the simplest way to access this data be to use ASSOC to associate a huge array of 2-byte variables with the file and then subscript it as necessary to read in either frames or groups of individual pixels.

Struan

---

Subject: Re: point_lun is slow
Posted by <karri on Thu, 28 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

George,

I am using FibreChannel disks and found out that reading a sector at a time feels like reading a floppy. With chunk sizes of 256K I get about 7MB/sec sustained speed. So at least on the Fibre you have to use large chunks to get any kind of performance.

I also tried 64K chunks but the performance was much worse.

On Wed, 27 Oct 1999, George McCabe wrote:
> do you have a rule of thumb for optimizing chunk size?

--
Regards,

Karri Kaksonen
Picker Nordstar

---

Subject: Re: point_lun is slow
Posted by korpela on Fri, 29 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <381860B1.7145@gsfc.nasa.gov>,
George McCabe <george.mccabe@gsfc.nasa.gov> wrote:
> Struan Gray wrote:
>
>> simplest way to access this data be to use ASSOC to associate a huge
>> array of 2-byte variables with the file and then subscript it as
>> necessary to read in either frames or groups of individual pixels.

>
>
> Struan,
>
> ASSOC - certainly the simplest and also the fastest in cases where every
> element of the data is needed.  but when for example only the
> [1538,395778,790018,... 434454018] location elements are needed from the
> file it is faster to point_lun to each.  furthermore, the size of each
> associated read can affect the overall speed of a procedure. it's a
> better understanding of this subtlety that i am hoping for.

In this case memory mapping is definitely the way to go.  Only accessed
pages get actually get read.

Eric

--
Eric Korpela | An object at rest can never be
korpela@ssl.berkeley.edu | stopped.
<a href="http://sag-www.ssl.berkeley.edu/~korpela">Click for home page.</a>

---

Subject: Re: point_lun is slow
Posted by korpela on Fri, 29 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

In article <38170AD4.2EC9@gsfc.nasa.gov>,
George McCabe  <george.mccabe@gsfc.nasa.gov> wrote:
> hello,
>
> reading from a data file at regularly spaced byte locations, 2 bytes at
> a time using point_lun - my program is abnormally slow.  i don't have
> enough experience to guess whether the poor performance is inherent to
> point_lun & readu approach or if there are options which are affecting
> execution adversely.

I generally map the entire file into memory (see my web site).  As long
as it's on local storage its usually faster than reading it into memory,
especially if the file is larger than physical memory.

Currently only works on Unix systems, though.  Haven't done a windows
version.

Eric

--
Eric Korpela | An object at rest can never be

korpela@ssl.berkeley.edu | stopped.
<a href="http://sag-www.ssl.berkeley.edu/~korpela">Click for home page.</a>

---

Subject: Re: point_lun is slow
Posted by George McCabe on Mon, 01 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

hello and thanks for the reply Liam,

i reason that the record size has to be larger than the seperation
between consecutive extracted elements in order for there to be a gain
in performance. the number of reads is reduced. also reading
sequentially in a chunk is faster than seeking to each datum, an
additional gain.

let's say the ratio of data read to data extracted is 'R'.

(the time it takes to seek and read the extracted data only) divided by
(the time it takes to read sequentially and extract data) is 1 when R
is what value? my guess is 100. what are your thoughts?

george

In article <38187A7E.8E60823A@ssec.wisc.edu>,
Liam Gumley <Liam.Gumley@ssec.wisc.edu> wrote:
> The following pseudo-algorithm reads records (chunks) of data from the
> disk in sequential order. Only records that cover the specified read
> locations are actually read from disk. Each record is only read once.
>
> Sort the array of read locations from lowest to highest
> Set the record size to 512 bytes (you can experiment with record sizes)
> Set the old record number to -1
> Start a loop over the read locations
> For this read location, compute the record number in the file
> If the record number is different than the old record number
> Read the current record
> Set the old record number to the current record number
> End If
> For this read location, compute the byte offset within the record
> Extract data from the record at the byte offset
> End Loop
>
> This method should be just as efficient for small or large numbers of
> read locations.
>
> Cheers,
> Liam.

>
> --
> Liam E. Gumley
> Space Science and Engineering Center, UW-Madison
> http://cimss.ssec.wisc.edu/~gumley