
Subject: Re: array concatenation and optimization

Posted by [Pavel A. Romashkin](#) on Wed, 26 Sep 2001 21:51:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst wrote:

>
> I changed some of my IDL code that reads a huge binary file from the concat method to
> estimating the number of points, added some slop and then read it all in a chunk (or in your
> case, a line) at a time, counting the points as I went. Before returning I simply truncate the
> array. Sped up my code by at least an order of magnitude.

I have offered this more than once before. It is basically what Paul is suggesting, except I don't read the number of points. I create a buffer (array) with 1000-5000 rows (with any number of columns), then read in a loop, appending the buffer to Result. When READF, Unit, Buffer passes EOF (and causes IO_ERROR, of course, trying to read past the EOF) I redimension the buffer only once, using the value from (FSTATS(unit)).TRANSFER_COUNT. Then, I return to the beginning of last read and read the last buffer correctly.

In fact, you could even ignore redimensioning, close the file and check validity of Buffer and truncate it instead, since Buffer does by then have all data in it from the READF that caused the IO_ERROR.

This eliminates unnecessary reads from the file, as it gets read only once. You don't need to know the number of rows in it.

I never notice this procedure when it reads data on the fly, it is fast.

Although I don't seem to work on ASCII files that are over 50 Mb.

In case of numeric data mixed with strings, I found no faster way than to read it all into STRARR and then deal with it - it is faster to process string array located in RAM than trying to read incrementally from the hard drive. Disk I/O is about the slowest thing your computer can be doing, except maybe accessing a CD-ROM.

Cheers,

Pavel
