Subject: Re: Concatenating arrays - speed issues?
Posted by ben.bighair on Tue, 07 Jun 2011 17:24:14 GMT
View Forum Message <> Reply to Message

On 6/7/11 11:48 AM, Rob wrote:
> Hi
>
> This is a pretty basic question but I'm trying to speed up some code
> and at the minute it's quite ugly and does this:
>
> 1) Loop over input data
> 2) Perform some calculations based on input data which may or may not
> produce a result we want to use
> 3a) If a result is produced and it's the first time within the loop,
> create an array to hold it
> 3b) If a result is produced but it's not the first time within the
> loop, concatenate the result to the array that's already been created.
>
>
> I'm doing the concatenation with something like:
>
> array = [[array],value]
>
> Now this works fine but as we get more input data it seems the
> concatenation is becoming quite slow (presumably as the arrays are
> getting larger and larger).
>
> The alternative I guess would be to define an array at the start with
> some arbitrarily large size, subscript the values to it and then check
> at the end to trim empty elements but that doesn't seem much nicer and
> in this case estimating an arbitrary size isn't that straight-forward.
>
> Is there an "IDL way" way to do this?
>
> Cheers

Hi,

I highly recommend Mike Galloy's "collections" for efficient resizeable arrays. It sounds perfect for your needs.

http://michaelgalloy.com/

and

http://docs.idldev.com/idllib/

Cheers,
Ben