Subject: Re: array concatenation and optimization
Posted by Mark Hadfield on Thu, 27 Sep 2001 22:25:03 GMT
View Forum Message <> Reply to Message

From: "Craig Markwardt" <craigmnet@cow.physics.wisc.edu>
> ...
> My pet favorite is to read the file line by line, but grow the array
> in chunks. I usually grow it by powers of two until a certain limit.
> Example (not tested),

I built essentially the same logic into my MGH_Vector class, see

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mgh_ve ctor__define.pro

The data are stored in a pointer array which is initialised with spare capacity. Elements can be added one at a time; every time the capacity of the array is reached it is extended (which means it is replaced by a larger one). After some trial & error I set the initial size to 1000 & the resizing algorithm to

new_size = round(1.5*old_size) > (new_size+1000)

The advantage of doing this inside an object, of course, is that all the details can be hidden and forgotten about.

Performance is acceptable: creating the object, adding 10^6 items (5-char strings), retrieving them all and then destroying the object takes 20 s (Pentium III 800). This compares with about 3 s to do the same operations with a plain string array, using the same logic to extend the array when necessary. Either way, the time varies more-or-less linearly with the number of items to be processed.

The timing code is in

http://katipo.niwa.cri.nz/~hadfield/gust/software/idl/mgh_ex ample_container.pro

---
Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

--