Subject: Re: Random-access of List() progressively slower for static list
Posted by tom.grydeland on Mon, 05 May 2014 13:25:59 GMT
View Forum Message <> Reply to Message

On Monday, May 5, 2014 12:25:19 PM UTC, Helder wrote:
> In IDL you're looking at a "singly-linked list of pointers" (from http://www.exelisvis.com/docs/LIST.html).

Or even double-linked, according to Chris Torrence:

https://groups.google.com/d/msg/comp.lang.idl-pvwave/fVyCtoR jmoQ/iq3-xHrA368J

> According to Wiki, lists of this type have advantages and of course lots of disadvantages if compared to dynamic arrays: http://en.wikipedia.org/wiki/Linked_list#Linked_lists_vs._dynamic_arrays

I understand this. I suppose I'm just disappointed that IDL's implementation is so ... underwhelming, performance-wise. Since their List and Hash datatypes appear to aim at matching the lists and hashes of Python or Perl, they could very well have incorporated some aspects of dynamic arrays to make them more generally useful.

> The performance difference between using values.add and values[jj] might be a direct reference to the last element of the list, making therefore add faster. But that is just speculation.

[...]
> Append processing time = 0.998
> Insert at beginning processing time = 0.979
> Insert at end processing time = 29.334

So I think you can safely assume that the List implementation keeps track of the last element, in order to insert quickly at the end. Furthermore, insert at a numbered node evidently doesn't check if it'll be quicker to count from the end.

> I hope you'll find your answers in wiki or from the tests above. What I've learned (the hard way) is that lists in IDL should not be used for doing what arrays can do. They should only takeover when arrays don't work well (insert elements, changes of type or array extension).
>

My test was actually quite close to what I want to do. I want elements that are short lists with heterogenous elements. I want many of them, and I want to access them at random. I know that there are ways of doing it in IDL without waiting forever, but these ways are not as _nice_ as they could be, using List().

> Helder

Thanks, and best regards,

--Tom Grydeland, Norut AS
