Subject: Re: Performance of a loop
Posted by Paul van Delst on Mon, 22 Oct 2001 15:07:58 GMT
View Forum Message <> Reply to Message

Martin Downing wrote:
>
> Paul,
> Im curious, can you explain why your modification should run faster?

The array access indices are listed as: [i,j,k]

The original loop order was
>>> FOR i = init, limit, 1 DO BEGIN
>>> FOR j = init, limit, 1 DO BEGIN
>>> FOR k = init, limit, 1 DO BEGIN

I suggested changing it to:
>> FOR k = init, limit, 1 DO BEGIN
>> FOR j = init, limit, 1 DO BEGIN
>> FOR i = init, limit, 1 DO BEGIN

i.e. reversing the i and k looping. IDL is like fortran in that array numbers are stored continguously in the order of i->j->k (opposite to C) so by looping over k as the innerloop, the access speed may suffer in that rather than loading numbers from adjacent memory locations, jumps over the i and j dimension would be required to load the next k-dimensioned number. Depending on the size of the arrays, this could incolve a lot of memory copying/gymnastics == time hog. Others more knowledgable than me about hardware would now start talking about cache
lines, translation lookaside buffer misses and other computey-type esoterica.

The upshot: always try to access memory contiguously.

paulv

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul