
Subject: Re: Array searching efficiency

Posted by [David Fanning](#) on Fri, 11 Feb 2011 00:03:15 GMT

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

Matt Francis writes:

> I'm going through and trying to squeeze every last bit of optimum
> efficiency out of a code I've been working on. I have a small, very
> simple, problem that I'd appreciate some experienced input on.
>
> I have a time series of maps, but the time stamps for them are not
> always regular (some maps get missed, some are late etc). I need to
> (many millions of times..) find which two maps a given time sits
> between, then interpolate between the relevant two maps at some given
> location.
>
> The first step is to establish, for a given time, which two maps I
> need to interpolate between. I have the times for each map stored in a
> single array, in time order. If I can work out the indices in that
> array of the two map times then I'm done. This is a simple problem to
> solve in any number of ways, but the question is which is the fastest?
>
> I've come up with this:
>
> iup = (WHERE(times-time_now GT 0))[0]
> ilow = iup-1
>
> I'm not sure how WHERE works 'under the hood', but assuming that at
> some level it loops over the given array, then optimally once [times -
> time_now] becomes positive you would stop searching. Implementing that
> kind of algorithm in a WHILE loop is probably slower than using WHERE
> though.
>
> Any thoughts or suggestions?

I would be shocked if it wasn't several orders of magnitude faster
(for this many iterations) to Histogram your times array with some
appropriate bin size and then ask "which bin" your time_now
was in with Value_Locate.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

