
Subject: Re: Efficient pattern-matching in a large array
Posted by guillermo.castilla.ca on Tue, 16 Aug 2011 16:41:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

> Hint: use ULONG64 (8 byte) comparison:...

Lajos, this is an awesome trick, hats off!!

It took me a while to understand it, so assuming there are more people out there who are intrigued by this (and also for myself, to understand it at a later date), here's a layman explanation:

Chris has a large 1-D array of Byte type (i.e., numbers ranging from 0-255). He wants to find occurrences of a specific sequence (what he calls 'pattern') of eight numbers (in his case [234, 203, 138, 216, 21, 52, 117, 39]). It happens that ULONG64 numbers are stored as a sequence of eight Byte numbers (in his case his 8 byte sequence = 2843236008186268650). Therefore you can convert each sequence of eight consecutive numbers in the array to a ULONG64 number, and then the problem is reduced to finding occurrences of that number in the 8 shifted, reduced (by a factor of 1/8) ULONG64 versions of the original array (where the shift is performed using the offset argument in the FIX function). Brilliant!

Lajos, I was wondering if your solution can be generalized to patterns of any length, could it?

Guillermo
