

Re: Idea for ECMA/C# Standard – compile time hash for performance

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-04/msg01860.html>

- *From:* "Bruce Wood" <brucewood@xxxxxxxxxxx>
 - *Date:* 7 Apr 2005 10:19:56 -0700
-

Two reactions.

First, what is the nature of the keys you're using for lookup? Account numbers? Something else? You may be able to design a better data structure than a generic hash table to improve your speed.

Second, can you take advantage of multithreading and parallelism, and in so doing move to a multi-CPU box and take advantage of those multiple CPUs? Of course, it all depends upon the nature of your problem: some problems don't scale well to parallel computing; others do.

-
- *Follow-Ups:*
 - ◆ [**Re: Idea for ECMA/C# Standard – compile time hash for performance**](#)
◇ *From:* WXS

- *References:*
 - ◆ [**Idea for ECMA/C# Standard – compile time hash for performance**](#)
◇ *From:* wxs
 - ◆ [**Re: Idea for ECMA/C# Standard – compile time hash for performance**](#)
◇ *From:* Bruce Wood
 - ◆ [**Re: Idea for ECMA/C# Standard – compile time hash for performance**](#)
◇ *From:* WXS
 - ◆ [**Re: Idea for ECMA/C# Standard – compile time hash for performance**](#)
◇ *From:* WXS

- Prev by Date: [**Re: Evaluate true in C# using switch**](#)
- Next by Date: [**Re: Using arrays to display a date.**](#)
- Previous by thread: [**Re: Idea for ECMA/C# Standard – compile time hash for performance**](#)
- Next by thread: [**Re: Idea for ECMA/C# Standard – compile time hash for performance**](#)
- Index(es):
 - ◆ [**Date**](#)

Re: Idea for ECMA/C# Standard – compile time hash for performance

◆ *Thread*