

Re: Hashtable faster than SQL ?!!

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-05/msg04124.html>

- *From:* "Dmytro Lapshyn [MVP]" <x-code@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 23 May 2005 13:50:55 +0300
-

Hi Steven,

There's nothing surprising. The Hashtable is (well, let's assume so) in the RAM, and its searching algorithm is optimized for performance. The SQL Server keeps indexes on disk, so even if the algorithm is similar, loading the indexes from disk takes time (as well as transmitting the result set over the network or over a named pipe).

--
Sincerely,
Dmytro Lapshyn [Visual Developer - Visual C# MVP]

"Steven Wolf" <apoc69@xxxxxxx> wrote in message
<news:5167D5D4-DDA4-4B9A-8419-8066DBE78C47@xxxxxxxxxxxxxxxxxxxx>

Hi guys,

can a hashtable be faster than a sql server??
i made my own O/R mapping with an "entity cache" (caching all mapped objects). So, when i get some data from my sql server, i map that data to an entity-object and i leave a copy of it in the entity cache.

the entity cache is a simple hashtable, which stores as key: the ID of the entity (the same id as the primary key of the sql-table) and the entity type.. and as value: the object itself.

```
so, when i access the cache, i say: Contact contact =  
EntityCache.Get( 5109,  
typeof( Contact ) )
```

i just made some tests and i filled the cache with 1 mio. contacts.

Re: Hashtable faster than SQL ?!!

the result: searching the hashtable for one value, costs mostly ~0.000.007 sec! (nanoseconds).. thats amazing!! my sql server can never reach such values! my dev. machine is just an amd 2 ghz with 1 gb ram.

of course i can search in the cache only by id, but it is the same way as the sql server would search for an index, isn't it? i thought about, to setup an extra caching-server... just read a big sql-table and store everything in the entity-cache..(updating/inserting/deleting or searching by other values is another topic).

i cant believe it!

steven.