

Re: File Searching, how to speed it up?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.performance/2007-05/msg00032.htm>

- *From:* "Robert" <no@xxxxxxx>
 - *Date:* Sun, 27 May 2007 20:09:42 +0800
-

Robert wrote:

I have a win32 C# app that needs to recursively search for a particular file type really fast. It always searches the same place for these files.

I'm using `Directory.GetDirectories` and `Directory.GetFiles` to do it currently. Because I know the location is the same every time I build a cache of all the files and directories. If a directory changes I update its contents on startup of the app.

Unfortunately when I do a last-date-of-modification check on a directory it only reports on changes of files that are directly under it. So I have to check every single directly rather than simply checking the parent directory.

While I get a significant improvement when none of the directories change (its much faster to load a file with all the entries in it then to do a directory search). In particular the first time the app loads up and has to search every file (around 100,000 and growing) which is painfully slow (2 or 3 minutes to start up the app).

What techniques/APIs can I use to speed up file searching?

Re: File Searching, how to speed it up?

Is there a way to more directly look at the raw data that make up the file-tables?

Change your filesystem is the easy answer. FAT32 is 2x-8x faster than ntfs.

From my generic WD 250gb HD with a 32gb partition:

64k Clusters NTFS Defrag=0 08m 22.31 .jpeg
64k Clusters NTFS Defrag=1 08m 11.82 .jpeg
Default Clusters NTFS Defrag=0 19m 23.75 .jpeg
Default Clusters NTFS Defrag=1 05m 44.15 .jpeg
FAT32 Defrag=0 02m 48.25 .jpeg
FAT32 Defrag=1 02m 42.78 .jpeg

If you have more data than FAT32 can hold, use ntfs with a larger cluster size.

64k clusters have a much better worst case, but a lower best case..

The above results were with 100,000 files of 275 kb each. Each of the results above was for 10 runs, with the filecache cleared between runs.

last, turning off 8.3 filenames helps a bit.

Thanks for your comments and effort put into this! Its very interesting that FAT32 would be faster.

This was paid research for a client.. The results were the jpeg's on my harddrive.

The filenames are informative however.

NTFS has unicode file names (2x the size) and a bunch of metadata. Something on the order of 1K (yes K) per file.. It also does security checks. All the above times were on xp64 as admin with ownership of the files.

FAT32 is old, and was designed for old slow computers with small HD's. NTFS was designed for servers, security, and file safety(journals, etc)

Unfortunately I can't change the the file system (unless it can be done per folder). If possible I would like the program to start up more quickly then 2minutes. Note that I don't need to read each file, I just need their names.

Re: File Searching, how to speed it up?

File system is per partition. So make a partition for your data..
Or buy another drive. They are cheap.

The above times were for simply scanning the dir's and building a list of all the files.

I'm think I may have to run this check in a separate thread, and have the files appear slowly. However I would rather have all files available as quickly as possible. Is there a async mode like you can do when loading files?

A background thread might help. User could do something with the partial list of files.

With async you would have to be careful not to overwhelm the disks. If the average disk queue length goes over 2 it does terrible things to SQL server performance.. On the other hand, if you have server drives (SCSI/SAS or even SATA with NCQ) you might get a doubling of performance due to less disk head movement (elevator seeks) if you are doing random IO.

Defragging does help.

Also where all the files in the same folder or nested under that folder?

Nested, 120 files per folder, 60 of those folders for each parent

Is it possible to defrag a particular folder?

No.

Have you tried it with windows indexing?

This is for the contents of files, so I do not think this would help.

good luck.

.