

Re: FAT32 or NTFS?

Source:

<http://www.tech-archive.net/Archive/WinXP/microsoft.public.windowsxp.general/2004-05/14713.html>

From: Al Dykes (adykes_at_panix.com)

Date: 05/14/04

Date: 14 May 2004 16:40:12 -0400

In article <eok7a0dfs9jmq96bh0s9psbo2hsaqejk8c@4ax.com>, cquirke (MVP Win9x) <cquirkenews@removethis.mvps.org> wrote:
>On Wed, 12 May 2004 19:42:19 +0530, "DILIP" <dilipr@#*&!%!.com> wrote:
>
>>> Well, do the maths. If files are 150k, and cluster space is 32k
>>> rather than 4k, then the biggest wastage would be $32k - 4k = 28k$ per
>>> file. Assume actual wastage is half that, i.e. 14k, per file. That
>>> boils down to under 10%; significant, but not carnage.
>>
>>10% wastage on 10 volumes is one whole hard disk wasted. Significant? Hell
>>yes, it increases costs! You should've known that considering it's the job
>>you do.
>
>That's worst-case. And if the files are 2M, you'd have to use 100 HDs
>to need the extra one, which really is a shrug.
>
>>> >Other considerations of the original poster were "resilience, recovery,
>>> >ease of repair".
>
>>> Yep – and once again, NTFS isn't quite what it seems.
>
>>> Read up on
>>> the shuffling NTFS does when it fills up... I can see the same sort of
>>> widening of critical windows there, as it re-balances between MFT and
>>> data space. Problems may also arise when software that assumes
>>> limitless memry (as bounded only by free swap space) hits the wall.
>
>>I know about the shuffling it performs. It is not as terrible as you make
>>it seem. For the benefit of the others reading this thread – Basically 12%
>>of the initial part of the HD is set aside initially (saving the 16
>>metafiles) for the MFT to prevent fragmentation (it's not even used at this
>>point of time). As and when data is required, the MFT is adjusted. Nothing
>>very "problematic" or sinister in there.
>
>And if the MFT is caught halfway through this process?
>
>Perhaps I don't see the bad mileage if FATxx that you do, because I

>don't use one big C: for everything. All of my FATxx-based systems
>have C: with 4k clusters, limited to just under 8G, and that's where
>most of the write traffic is going on. The bulk of the capacity in E:
>is FAT32 with big clusters, but there's not much traffic there.
>
>I'm certainly not recommending 120G HDs be set up as one big FAT32
>volume; even though that still has maintainability advantages over
>NTFS, the reliability gap may be as wide as you claim.
>
>There is one factor that could lead to software crashes, and that is
>uncertainty about free space. An app may query the system for free
>space, be told there's enough, and then dump on the HD without
>checking for success. Normally it would be concurrant traffic and
>disk compression that would cause this "oops", but FAT32 (not FAT16)
>does add an extra factor; the free space value that is buffered in the
>volume's boot record, which so often gets bent after a bad exit.
>
>Then again, AFAIK startup always checks and recalculates this value;
>it's one of the extra overheads of FAT32 at boot time.
>
>>I have had an NTFS volume with as less as 25MB free out of 10GB
>>running with absolutely no problems over a long period of time.
>
>Well I should hope so, as that's the mileage I'd expect in FATxx as
>well. It shouldn't blink even if free space fell to 5M. C: might
>look like it blew up with 25M to go, but that's prolly because a few
>seconds ago it may have been down to 0k free due to a temp or swap
>splurge... then again, I've seen 0k-free C: and I haven't seen data
>carnage. The data carnage I see is usually where RAM has been flaky
>for some time, or there's been a malware strike, or the HD is failing,
>or the PC was overclocked, etc.
>
>FATxx doesn't just fall over for no good reason, from what I've seen,
>though persistant issues from bad exits might compound into
>cross-links later. Not sure if that does happen, but possible, though
>I'd expect to see a lot more cross-links if that were the case.
>
>>> >In such situations, it becomes tricky to save the OS, and depending on
>>> >the seriousness of the crash, data loss possibilities are considerable.
>
>>> I've done a lot of data recovery on FATxx, and that's the point; it is
>>> possible to recover lost data in such cases.
>
>>Lost data OK. How about corrupt data.
>
>Depends what has gone wrong. Corrupt data can be better than none,
>especially where text is concerned; OTOH, half a .DLL is no bread :-)

>It's important to know what is corrupted and what is not – and that is
>what I have against "auto-fixing" junk. It's the equivalent of
>throwing the needles back into the haystack.

>

It's statements like this that lead me to believe there is no experience with business data, here. You can't tell the auditor that "you can only recover part of the database". It's everying, or I'm fired. I don't know how I'd figure out what the correct parts of a multiMB TIFF file are. I have visions of printing each cluster out on my color printer, individually, cutting them out, spreading them on the floor, and treat them like a jigsaw puzzle. Recovery of all important data is based on backups, which are certain, dtat recovery is not guaranteed to get your work back under all types of equipment failure.

Jeesh. a decade of experience with thousands of NTFS-based systems at one of the biggest banks in the world must count for something. Every time I looked at an old machine that was still FAT32 I found all these mystery .CHK files. These don't happen, all practical pruposes with NTFS. If one of my tech staff raised his hand and said "I can might be able to recover those files for you sir, but it will take me a day, and it may not get the data back", I'd tell him he had more important things to do.

I also don't understand what data you say NTFS is deleting as part of a tranaction rollback. The blocks were in an undefined state, so I down't _want to_ recover them.

>>> >recovery and ease of repair --- NTFS file systems don't need a scandisk
>>> >add on per se, simply because they don't need one. Data verification
>>> >technologies imbedded in the file system ensure that data is only written
>>> >on to the disk, when it is verifiable. If it cannot be read back, the
>>> >transation is simply rolled back.

>

>>> *Read: The damaged data is stimply and irreversably destroyed.*

>

>>> *The data remains intact. I have had power failures while working with
>>> files and defragging. The files remain - albeit at the cost of the
>>> changes. No issues there. I don't know what and why you're trying to
>>> project this wrongly.*

>

>>> *What happened to the incomplete transaction? That's the data I want
>>> back. I don't want some clueless fixer deciding for me that it's
>>> corrupted and therefore should be discarded without a trace.*

>

>>> *It's very easy to look bulletproof if you simply destroy everything
>>> that may be damaged. You could do that in FATxx as well - in fact,
>>> the duhfault behaviour is close - simply by maintaining a list of
>>> files open for writes, which are automatically deleted on bootup.*

>

>>> *But that is not data preservation.*

>

>>> *This is what I mean; NTFS "doesn't need" Scandisk simply because it
>>> behaves as if Scandisk was running all the time, with the most*

microsoft.public.windowsxp.general: Re: FAT32 or NTFS?

>>> *ruthless trade-off between your data and file system integrity.*

>

>>*I don't understand this statement. Cut the flowery language and start explaining.*

>

>*Throwing away any transaction that is not complete is not data preservation. Basically, you get the worst-case auto-fixing Scandisk result, i.e. as if you'd let Scandisk automatically fix all errors and throw away all lost cluster chains.*

>

>*All of this nonsense about transaction rollback "doing away with the need" for disk maintenance utilities hinges on the only problem being the interruption of sane file operations.*

>

>*What about damage from other causes, such as wild disk writes, bad RAM corrupting content and address of writes, program crashes, and deliberate malware raw disk writes as per Witty?*

>

Backup procedures, tailored to fit a business risk assesment cover everything.

--

Al Dykes

adykes at p a n i x . c o m