Re: memory reading and writing

Source: http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2004−07/0819.html

---

From: Beth (*BethStone21_at_hotmail.NOSPICEDHAM.com*)
Date: 07/19/04

Date: Mon, 19 Jul 2004 16:03:31 GMT

Robert Redelmeier wrote:
> *Gyps wrote:*
> > *Haven't all sorts of user−mode services started by the NT*
> > *kernel by the time login prompt is shown? All that code*
> > *wouldn't be able to fit my 16KB L1 cache...??*
> >
> *It doesn't need to! All that needs to fit is the important*
> *loops. Execute−once code doesn't benefit from cache other*
> *than the "burst−in" instruction loading. Just as important*
> *is that data cache, particularly the stack.*

Yes; And, of course, during the boot process, it's mostly "initialisation" and "run once" code...so, the caches ain't really helping much with that kind of thing, anyway...

Plus, remember that Windows uses "on demand" loading...hence, an application or service might be "loaded" but this doesn't mean the entire file is actually in memory...indeed, if not in use, then − as we know from the other thread around here − it'll likely be _blocked_...

This conspires to mean that once initialisation is run (which is "once only" stuff that does not benefit much from the caches, anyway), it'll fall into its main loop and probably block...very little of the actual service / application needs to be actually directly held in memory or cache directly...in addition to the fact that, to speed up the booting, Windows does do a lot of "avoiding" to load things until strictly necessary...

In truth, what you should perhaps do is set up an XP system with one user account and no password so as to make it by−pass the "Welcome" screen and log−in directly...also, make "Task Manager" one of your startup programs (put a link in the "startup" folder or registry entries or whatever) and, just before the test, leave Task Manager open on the "performance" tab (it remembers which tab you were last looking at next time you open it so that the idea is that Task Manager will appear straight away and show the system performance)...we need to add this − it won't skew

the results too much in a comparative sense, if you have it load with and without caches − because Windows loads things while log ging a user in but there's no clear "dividing line" you could normally see to know when to stop your stopwatch...but looking at the "performance" in Task Manager, it's truly completely finished the _full_ booting process when it all settles down and the "performance" drops to 0% ("system idle process" is 99% and everything else is 0%)...and that's where you'd hit "stop" on your watch ;)...

Again, to make it a fair test too, then we'd really need a brand newly installed version of the OS so that there are no extra "tray icons" beyond what is part and parcel of the OS itself...

Ah, it's tricky to get it exact...but, indeed, up to the "welcome" screen might not really be representative, as Windows still takes some time to get from that screen to actually being fully finished initialising after logging a user in...as mentioned, it's an "optical illusion" kind of thing: It looks like Windows has finished booting...

Actually − wow! − you know what I'm thinking? Microsoft have missed a trick...while displaying the "welcome" screen, it can actually "cheat" at that point: While waiting for the user to choose an icon and type in their password, Windows can "background load" the rest...hey, if anyone's writing their own OS out there, consider that as a neat "cheat"...have some elaborate "welcome" screen which animates in and then the user has to select their user account and type in the password...but the boot process can actually still be _on−going_ in the background continuing to load the basic services...

"Distraction"; The oldest slight−of−hand trick in the book...it's actually surprising Microsoft − the Kings of "make it _look_ good then no−one will notice how crap it is under those looks" − didn't think of it...

And, I may be wrong here as I'm just going by "observation"...but I've noticed that XP boots slower when it wasn't shutdown properly or when you make a big settings change...some process starts up that's actually quite lengthy in the background...now, part of that might be a touch of "invisible NTFS recovery" stuff, granted...

But I'm thinking there's also a possibility here that what XP might be doing to help speed up the booting process is a similar trick to "hibernation"...that is, if the process booted up okay last time and none of the settings have changed, then it can just save the _memory image_ of the processes as a set of straight disk files in a "boot cache", just after booting that process properly...then on a subsequent boot, you can literally

just load up the memory image straight from disk and not bother to redo all the "initialisation" all over again (it's just one literal continuous read from disk: Just dump the file into memory "as is"...rather than load this, initialise that, load that DLL, initialise that, etc. ;)...but, yeah, if the settings change or it doesn't shutdown properly (for "safety's sake" :) then you can't use that memory image and just discard it, booting up and initialising normally (takes a little longer but most boots when settings aren't any different from last time and it "shutdown" properly (for "safety", the "cache" should be discarded in case the memory images aren't actually trustworthy) should be able to load quicker :)...

This'll improve the booting speed in the majority case...indeed, it's a bit like those old "Action Replay" cartridges for older machines where you press a button and then just dump the memory image in a compressed format...it loads quicker and you could "save" a program practically anywhere in its execution (like "saving" a game on "level 5" with full "lives" because of how tricky it was to get there without losing a life...hey, what can I say? I'm not very good at computer games usually...I _need_ the assistance like that! ;)...

Again, if Microsoft ain't doing it, then perhaps they ought to consider it...couple this with the idea of "interleaving" the CPU−based parts with the I/O−based loading parts (as the disk I/O loading the next driver or whatever can be done _in parallel_ to the CPU initialising the current driver or service or whatever :)...use disk compression on the files...and, well, _OPTIMISE_ the size of these basic OS services...put all that together and you _could_ have one highly impressive boot−up speed (at least in "perception" terms ;)...

Mind you, older machines used to keep the OS in ROM and the typical boot time required? Ummm, practically nothing...appears on the screen almost instantly...and the limited space and cost of adding on ROM chips would _force_ someone like Microsoft to pay more attention to optimising to fit it all into affordable ROM chips...although, ummm, perhaps we shouldn't talk in these terms: Windows on ROM as part of the PC itself? Then it would _literally_ be impossible to escape!! Nah, perhaps _NOT_, on second thoughts...let's Hope Bill wasn't listening then and started "getting ideas" about how to really monopolise and eradicate the competition: Windows on ROM? All manufacturers would probably succumb and install it...then the option to boot up an "alternative" could completely vanish, as it boots from the ROM chips automatically...also something "open source" can't really follow without money getting involved...*gulp* I never said it, you never heard it...don't let this get anywhere near Microsoft's ears! ;)...

Beth :)