Whose FAULT Is It, Anyway?

Way back in the 1980s, I was introduced to Moore's Law[1] in an engineering class. It says that computing power doubles approximately every 18 months to two years. This same law appears to apply to many computer-related items such as processing speed, memory capacity, and even digital camera resolution. There's a lot of similar laws for disk size, power consumption, network capacity, etc. Moore's Law has held true since its origination in 1965, and will probably hold true for at least the next decade. Every few years engineers will say that we've reached the limit of Moore's Law, but new technology keeps proving it true. The bottom line is that technology grows at an exponential rate.

My first computer, a Commodore SuperPet that I bought back in 1982, had a whopping 128 kilobytes (KB) of memory. I don't recall the clock speed, but it was a relatively slow 6502 processor that I believe was at about 1 megahertz. (As an historical point, the Commodore SuperPet *also* had a 6809 processor, and you could run dual operating systems and interpreters for Pascal, APL, FORTRAN, COBOL, plus the obligatory BASIC interpreters).

Twenty-six years later, my current laptop has three gigabytes (GBs) of memory. This pretty much follows Moore's Law, as does the processing speed of my current machine, a two gigahertz (dual processor)[2].

It appears that Moore's Law also applies to the size of the operating system. Remember MS-DOS 2.11? Back in 1983, it loaded in 64KB—and left you room to run your programs! Windows 95 (12 years later) took 50 megabytes (MB) of disk space. And now with Windows Vista, Microsoft says you need 15GB of free disk space and 512MB of memory (still following Moore's Law).

What's the point, you ask? I'm *not* Microsoft bashing. I *like* my operating system. I find it useful to run multiple applications, tons of sidebar gadgets, high-resolution graphics, and have music playing in the background.

However, with the increased program size, did you know that the chance for failure goes up, too? I know that Vista is a pretty solid operating system. I haven't had a single blue screen of death, and only about three needed hard shutdown and reboot occasions since I bought it last year (as opposed to about three-a-day from my first experiences with Windows 95, if I recall). I'm talking about the chance of failure that comes from large-scale reliance on the compliance of others. And reliance on the rapidly expanding technology raises the potential for problems.

In our office, we have a one terabyte (TB) four-disk RAID (redundant array of inexpensive disks, as named by the inventor, or occasionally known as redundant array of *independent* disks)[3] cluster. It has the very reasonable name of "Terrabyte[4]." If any two of the four-disk cluster fails, we still have a complete set of data. Until an "unexpected" failure took it totally down. Our domain name system server died—and with it down, Terrabyte was unable to grab an address. It took us a bit of time to locate the problem, and figure out how to reconfigure it. Of course, eventually, we realized that we could just plug directly into a computer. Then we realized that the permissions on the file access were based on domain authentication; so even though I could plug the device directly into my computer, it couldn't authenticate the access. Sure, it was fixable, but the delay cost several of us a bit of work. And, I admit, there was a bit of momentary panic when somebody asked, "Just in case, we *do* have a backup of it, don't we?"

We all have become dependent upon the increasing complexity of new technology. And when the technology fails, we all feel powerless. It's not like any of us can keep four or five different backups around on floppy anymore—backing up a TB RAID cluster takes some serious storage!

The point is that increased power, increased memory, and increased disk storage bring increased PPoF (Potential Points of Failure)[5]. And you need to plan for these failures.

Are you developing large-scale applications? Have you considered what to do in case the network fails? The database fails? How many backups do you have? Where are the backups located—having them in the same location really won't help in case of fire or flood, will it? Whatever technology you implement, eventually one of your users will run into a case where something goes bad, and they are going to expect you to have thought of the potential problem, and developed a contingency plan for it!

Technology lures you in—like when you're stuck in the airport, flight cancelled, you need to re-book, and you realize your cell phone is out of juice. Backup? Tried to find a pay phone lately? Kind of makes you long for the days when a spare deck of cards in your desk took care of your backup needs.

Speaking of faults, this column was almost late because the e-mail from the CrossTalk editors reminding me my article was due was somehow misdirected into my junk mail folder. I hesitate to state how great my life would be if the other 99 percent of my daily e-mail was similarly (but faultily) misdirected. If only Outlook had an "I.Q. filter," similar to caller ID. Then, when folks complained that I never responded to their e-mail, I could say "Honest, it's not *my* fault!"

—**David A. Cook, Ph.D.**
The AEgis Technologies Group, Inc.
dcook@aegistg.com

Notes

1. Wikipedia. <http://en.wikipedia.org/wiki/Moore%27s_ law>. And before anybody corrects me, yes, I know that Moore's Law originally referred to the number of transistors on a chip.
2. See <http://nano-taiwan.sinica.edu.tw/2008_WinterSchool /index/Moore%27slaw%20graph2.gif> for an image of the growth in Intel Processors.
3. Wikipedia again.
4. Why the extra "r"? Because my granddaughter is named Terra, I love her, and my office was foolish enough to let me name our RAID cluster.
5. Yes, I made this one up!

Can You BACKTALK?

Here is your chance to make your point without your boss censoring your writing. In addition to accepting articles that relate to software engineering for publication in CrossTalk, we also accept articles for the BackTalk column. These articles should provide a concise, clever, humorous, and insightful perspective on the software engineering profession or industry or a portion of it. Your BackTalk article should be entertaining and clever or original in concept, design, or delivery, and should not exceed 750 words.

For more information on how to submit your BackTalk article, go to <www.stsc.hill.af.mil>.