Throughput and Latency

From iGeek

Jump to: navigation, search

(/File:FireHose.jpeg)

Speed is relative. What kind of speed are you talking about? There are a few aspects to speed on a computer. People start looking at benchmarks, or hearing numbers, and they don't always understand what they mean -- or what changes will mean to them. For example if a machine is twice as fast as another, why doesn't it take half as long to do something? Well, mainly because computers are complex systems, and mostly because twice as fast at one thing is not twice as fast overall. Lets start with the basics, the difference between throughput and latency.

Types of Speed

The first thing to know is there are two types of "speed" for computers. There is ***throughput*** (thru-put) and ***latency***.

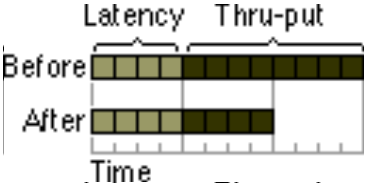
Latency is how long until something starts. Think of "reaction time". Imagine a race. Now the referee fires the starting gun, how long until the runners react to the sound (and start running); that is the latency (delay). Throughput is how fast they run, once they get up to top speed. Or another common example is a fire hose; turn the fire hose on pointing at a "friend". It takes a while for the water to get to them, that's the latency; how much water they are hit with (and how fast it is going) is the thruput. A garden hose may have less latency than a fire hose (probably because of length), but in thruput there's no contest.

Internet

On the Internet the results are very obvious. Click on a link (or type a URL) and there will normally be a one second (or more) pause, while the network does a DNS lookup (the computer/network figures out who you want to talk to talk to). This delay is your "latency". Then after it figures out what site you want to talk to (what page you want to load), it takes a while to send the data across. That transfer time is based on the thru-put. And after that there may be a bit more latency that is based on how long it takes your computer (Browser) to display the information it just got across the net. But let's just ignore that for now for the sake of simplicity.

This is why that when you get a modem that is 2 or 4 times faster; your Internet experience may not be that much faster. The modems thru-put (speed at which it can transfer data) HAS gone up, often by many times, but the latency of the Internet itself (and the speed of the servers) has not changed.

Look at the following example:



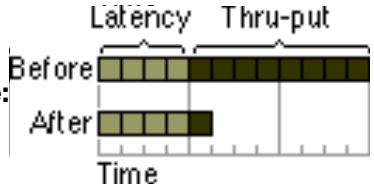
(/File:LatencyAndThruput.1.gif)

If it took 12 seconds to do something (say load a page), and 4 seconds was latency and 8 seconds was because of thru-put, then doubling thru-put (the modem speed) would **not** halve the time it takes to load that page (like most people would think). In fact, it takes 2/3 of the time that it took before (8 seconds) not 1/2 (6 seconds) that people assume.

The thru-put (modem) **is twice as fast**, and there is not any false advertising -- but that doesn't usually mean the real world results are twice as fast. Your browsing experience is not based only on the modem speed -- but on the speed of the ISP you are using, how fast the network you are on can lookup sites for you, how fast your computer can process and display the data, and so on.

That was a mild example of the problem.

Imagine a modem manufacturer claimed an 8 times performance increase. I can see the advertisements now, "New super-sportster-ultra-turbo-hyperdrive-mode, up to 8 times faster".

Back to our example: 

(/File:LatencyAndThruput.2.gif)

Sure enough, the thru-put (modem) is 8 times faster at transferring data than before and only takes only 1 second instead of 8 -- wow. Of course our fixed time (latency) is still four seconds, for a total of 5 seconds. The real world results are that the new modem only makes our browsing experience barely over twice as fast.

We can keep going. If we made the thru-put 8 times faster again (64 times faster overall), it would take 4.125 seconds to get the same data across -- or a barely noticeable difference at all. So you can see there are diminishing returns on speed, depending on what you are speeding up.

It is all a game of ratios and what you are doing. If you are transferring a file that will take 2 hours, then you don't care about latency and 5 or 10 seconds either way on that since the latency is such an insignificant amount of time - you care about throughput. Yet, if you are transferring 100 small files for a single page, and the latency is 4 seconds for each file (and the throughput is less than 1 second), then it is the latency that is torturing your performance and making your throw things at the screen.

PCI Bus

Those examples were using Internet and modem type speeds, where you can see both the latency and thru-put. But there are items on your machine that have the same problems, but are far far faster. Imagine we are talking about the lower levels of a computer.

Let's talk about the bus speed on a PCI card. Let's say it takes 10 cycles to "arbitrate" and setup a time to use the bus (to talk to another card). Once that latency (arbitration) is handled, the bus can send 4 bytes of data for each cycle after that. In order to send only one byte of data to a card it takes 11 cycles. But to send 1000 bytes of data it does **not** take 11,000 cycles, it takes only 1010 cycles; PCI is basically ten times as efficient at sending large blocks of data as it is at sending small ones.

> But it is not linear growth -- PCI isn't significantly more efficient at sending 100 bytes, or a 1,000,000. As long as your packets are like over 100 bytes at a time, you aren't like to see very substantial difference by doubling or quadrupling data (packet) sizes.

Once again your performance (speed) is actually based on what you are doing. PCI bus can do 33 Million cycles in a second (a lot of transfers) -- this is so much data transfers (and so quickly) that we really only measure statistically. Yet you could get a dramatically different result depending on if you were doing 3 million transactions of 1 byte each, or one transaction of 132 Million bytes of data. (Or a difference of up to 45 times).

> Fortunately, most data moved across PCI are packets of dozens of bytes -- so the latency overhead doesn't hurt much. So this is only a theoretical example.

Drives

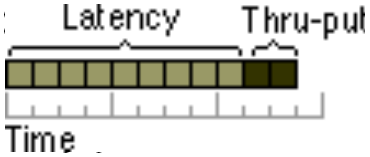
Drives are like a record player, for those that remember what that is (a spinning piece of vinyl that they used to put old music on). There are a few types of latency on drive specs. One latency is how fast the little disk is spinning and where you are on the disk, and how long it takes for the platter to rotate until the data you want under the head (the head is like the needle on a record player). Another drive latency can be measured on how long it takes for the little head to get from one edge (track) of the spinning platter, to the other edge (or another track) -- because you can't read the correct data until your are on the right track (this is the equivalent of lifting the arm/needle and changing songs). Those are both types of latency -- how long it is going to take until you can start transferring the data you want. Then there is the thru-put -- how fast the data will transfer once you get there.

Drive specs (and most performance specs) are very complex because there are a thousand different ways to measure that performance. Do you measure latency as track-to-track -- how long it takes for the head to go from one track to the one right next to it (assuming that the data will stack up neatly and the data is unfragmented) -- or do you assume worst case and measure latency as how long it takes to go back and forth between the two furthest tracks? There is also how long it takes for the platter to spin until the data is underneath (which is additional time, and can be a long time for a drive/computer, even when the platter is spinning at 7200 RPM's). Then there are many different ways to measure thru-put as well -- best case, worst case, and so on. All these factors make it hard to figure out how fast something is -- and of course all manufacturers love to publish the specs that they are best at (and often hide or ignore the ones that they are not good at).

Many people now just "profile" drives. They build up a huge set of averages, working with lots of little files, lots of big ones, scattering files all over, then reading random files, and making a big average -- and trying to rate the drives that way. But that isn't that good a solution either, because if you are doing 90% Photoshop or video work (with huge files), then those factors/specs that average in small file transfers and low-latency issues are all irrelevant to you -- you care about the throughput! On the other hand, I'm doing compiles and programming, which often requires thousands of small files. I care about latency far far more than throughput. So all these low-level specifications help a few people understand what drive will be better for them (if you are enough of a techno-weenie to understand the specs) -- but just bedazzle and confuse the layperson.

Balance is the key to speed

Remember, making one thing faster may have little effect at all. Speed isn't about the biggest or smallest numbers -- speed is about learning to pick the **RIGHT** numbers to improve.

Remember this image: 

(/File:LatencyAndThruput.3.gif)

I could make the thru-put 10 times faster and the user would see a difference of only a few percent. But if I made the latency only 2 times faster, the user would notice almost all that speed benefit.

So it is **NOT** about the specs. Don't trust people that try to sell specs to you; they are often misleading. Users need to care about the real-world results. Companies often like to focus on the specs, and especially on the specs they do best, while ignoring what that means for the overall System. But things that sound great in theory often aren't in practice. One of the biggest problems is that PC users have often been deluded by visions of grandeur and specs. "8 times faster", -- while ignoring what they really mean (faster at what?).

Conclusion

This article only explains one small factor of speed but should give you an idea of how complex things can be. Imagine loading a file from disk; a computer has drive speed (throughput and latency) intermixed with processor speed, bus speed, memory speed, and dozens of other things (applications and drivers) that can all be going on at the same time, each with their own issues about throughput and latency. Any one of those can be a bottleneck (slowing point), and they can all interact. The specs of improving one thing or another is important to engineers, but users need to ignore them and focus on what it really means to them.

I hope this article gives you a little better understanding of speed and the issues of latency and thru-put, and a little bit more about general performance in Personal Computers. At least now you can understand why something can be rated as 10 times faster even when you see far more modest results. No one was lying, they really are making some parts that much faster, you just don't notice it because they didn't make the things you needed faster.

The trick of being an engineer/designer is learning how to balance many things (many types of performance) to get the best results for your users but you can't do that well unless you have control of the whole system, and it is harder to sell than more misleading specs. So when someone starts spouting specs, that's your cue to "Caveat Emptor"; let the buyer beware.

Tech (/Technology) | **Programming** : Anti-aliasing (/Anti-aliasing) • Basics of BASIC (/Basics_of_BASIC) • Big or Little Endian (/Big_or_Little_Endian) • Binary, OCTal, HEXadecimal (/Binary,_OCTal,_HEXadecimal) • Command Line Interface (/Command_Line_Interface) • Databases (/Databases) • Digitized Sound (/Digitized_Sound) • Enterprise Tools (/Enterprise_Tools) • FUD (/FUD) • Forward Compatibility (/Forward_Compatibility) • Free Features (/Free_Features) • Hack, Crack or Phreak (/Hack,_Crack_or_Phreak) • Hiring Programmers (/Hiring_Programmers) • History of Visual Basic (/History_of_Visual_Basic) • How does compression work? (/How_does_compression_work%3F) • MHz or GHz (/MHz_or_GHz) • RISC or CISC (/RISC_or_CISC) • Raster Images (/Raster_Images) • Software Consultants (/Software_Consultants) • Software Development Live Cycle (/Software_Development_Live_Cycle) • Synthesized Sound (/Synthesized_Sound) • UNIX (/UNIX) • What is MP3? (/What_is_MP3%3F) • What is a WebApp? (/What_is_a_WebApp%3F) • Why is software so buggy? (/Why_is_software_so_buggy%3F) •

Written 1998.09.17