Preshing on Programming

Feb 08, 2012

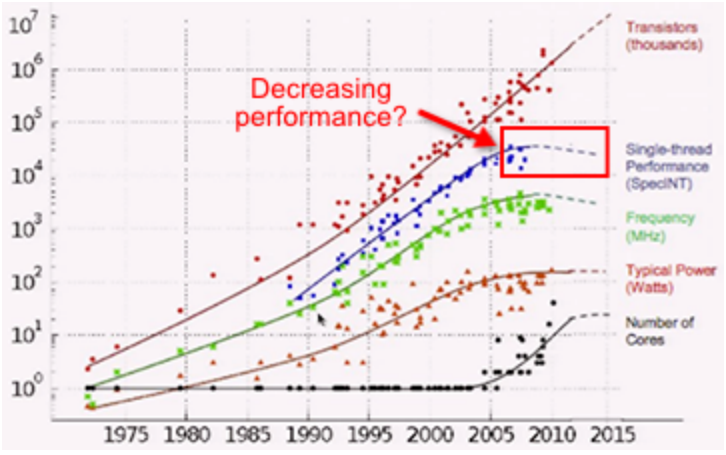
A Look Back at Single-Threaded CPU Performance

Throughout the 80's and 90's, CPUs were able to run virtually any kind of software twice as fast every 18-20 months. The rate of change was incredible. Your 486SX-16 was almost obsolete by the time you got it through the door. But eventually, at some point in the mid-2000's, progress slowed down considerably for single-threaded software – which was most software.

Perhaps the turning point came in May 2004, when Intel canceled its latest single-core development effort to focus on multicore designs. Later that year, Herb Sutter wrote his now-famous article, The Free Lunch Is Over. Not all software will run remarkably faster year-over-year anymore, he warned us. Concurrent software would continue its meteoric rise, but single-threaded software was about to get left in the dust.

So, what's happened since 2004? Clearly, multicore computing has become mainstream. Everybody acknowledges that single-threaded CPU performance no longer increases as quickly as it previously did – but at what rate is it *actually* increasing?

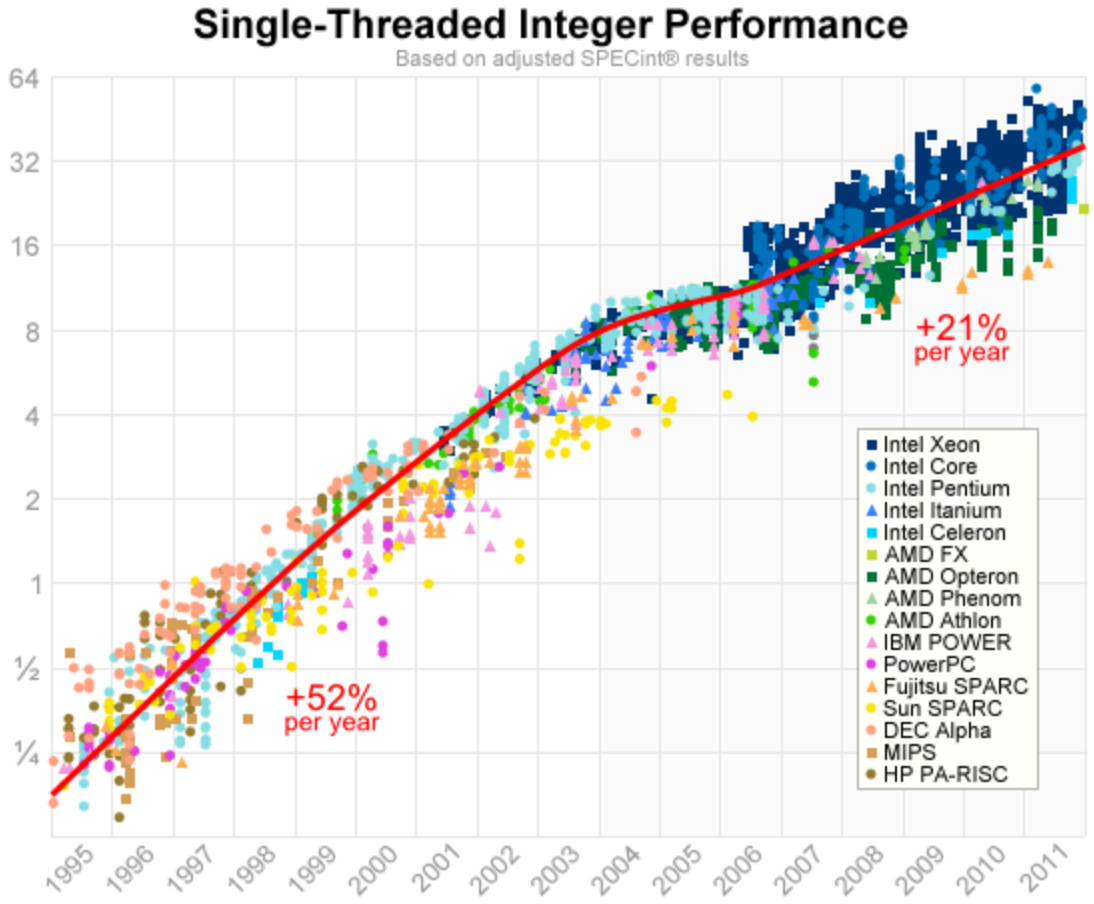
It's tough to find an answer. Bill Dally of nVidia threw out a few numbers in a recent presentation: He had predicted 19% per year, but says it's turned out closer to 5%. Last year, Chuck Moore of AMD presented this graph, suggesting that single-threaded CPU performance recently started going backwards:



These figures aren't really consistent, and both struck me as a little low. Moreover, I couldn't find another source to corroborate them. So I decided to crunch the numbers myself. I turned to SPEC, an industry-standard benchmark that's been going strong since 1989. It's the same benchmark used to plot a few data points on the above graph.

SPEC licenses their benchmarking software to various companies, collects results back from those licensees, and makes those results available on their website. One of their benchmark series, SPECint, was designed to measure the single-threaded integer performance of a machine. That sounds perfect, except for one catch: many licensees use automatic parallelization. I took some pains to remove those results from the dataset. I'll share the method at the end of this post, and you can let me know if you think it's valid.

I fetched SPEC's data on Feb. 7, grouped the results by CPU brand, and generated the following graph. It consists of 5052 test results from 715 different CPU models, all gathered over the last 17 years:



Each test result is plotted according its hardware availability date, and the vertical axis uses a logarithmic scale. The graph incorporates results from three different benchmark suites (CPU95, CPU2000 and CPU2006), but I've normalized the results in order to see historic trends.

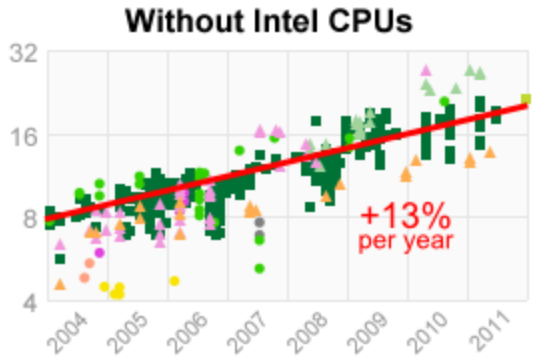
The red line is meant to represent **mainstream** CPU performance. I drew it manually, using the less-than-scientific method of eyeballing the points for Pentium, PowerPC, Athlon and Core. If you're willing to trust this line, it seems that in the eight years since January 2004, mainstream performance has increased by a factor of about **4.6x**, which works out to 21% per year. Compare that to the **28x** increase between 1996 and 2004! Things have really slowed down.

Here are a few machines located along the red line in the graph:

Hardware Availability	Adjusted Result	CPU Model	Clock Rate	CPU Cache
Feb 2004	8.1	Intel Pentium 4	3200 MHz	28KB L1, 1MB L2

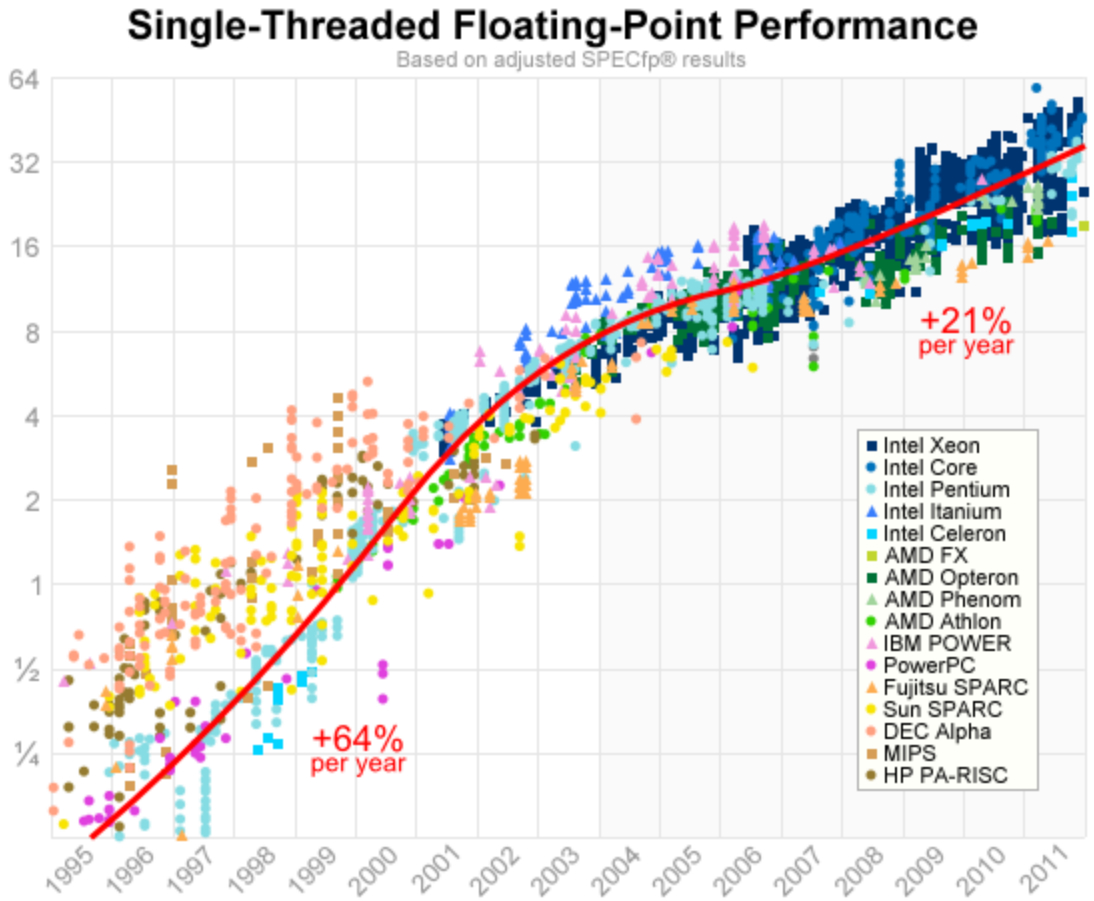
Hardware Availability	Adjusted Result	CPU Model	Clock Rate	CPU Cache
Jun 2005	10.5	AMD Athlon 64 FX-57	2800 MHz	128KB L1, 1MB L2
Jul 2006	11.4	Intel Core 2 Duo E6300	1867 MHz	64KB L1, 2MB L2
Jul 2007	13.3	Intel Core 2 Duo T7700	2400 MHz	64KB L1, 4MB L2
Sep 2008	17.9	Intel Core 2 Duo T9600	2800 MHz	64KB L1, 6MB L2
May 2009	21.8	Intel Core 2 Duo E7600	3066 MHz	64KB L1, 3MB L2
Jul 2010	24.3	Intel Core i3-540	3067 MHz	64KB L1, 256KB L2, 4MB L3
Jun 2011	31.7	Intel Pentium G850	2900 MHz	64KB L1, 256KB L2, 3MB L3

As you can see, Intel deserves credit for squeezing out the most single-threaded performance since 2004. If you remove all Intel CPUs from the data, a different picture emerges:



This is not too surprising, as AMD is pretty open about their stance on single-threaded performance. Bulldozer, their latest microarchitecture, is meant to shine in multithreaded workloads.

So far we've only looked at integer performance. SPEC also publishes SPECfp, an equivalent benchmark for floating-point performance. Floating-point performance has always been important for heavy-duty computation such as scientific simulation or 3D rendering. Here are the results, which I've also adjusted to eliminate autoparallelization:



Prior to 2004, it climbed even faster than integer performance, at 64% per year: a doubling period of 73 weeks. After that, it leveled off at the same 21% per year.

Up until 2002, we see a huge difference in floating-point performance between mainstream and workstation CPUs. The Alpha, SPARC and MIPS all ran up to 8x faster. Of course, you had to pay $10000 or more to get your hands on such a workstation. This is an interesting reminder that CPUs are, in fact, things created by businesses to make money! They don't become faster entirely by technological forces. They become faster by economic forces.

Which brings us back to the present day. For reasons which others understand better than me, involving thermal design power and ILP, it's now more cost-effective for manufacturers to pack additional cores onto a die than to push the single-threaded performance envelope much further.

Given the significance of this shift away from single-threaded performance, I was surprised to not find more information about the actual trajectory of performance since 2004. At the same time, I can't guarantee that the data I've presented perfectly reflects single-threaded CPU performance. I think my conclusions are fair, but any feedback or criticism about the approach is more than welcome.

How These Graphs Were Generated

All Python scripts are available on GitHub. These scripts will download, analyze and adjust SPEC's data, and render the graphs. If you'd like to run them yourself, see the README file for exact instructions.

As already mentioned, recent compilers like Intel C++ and IBM XL feature automatic parallelization, and it greatly skews the results towards certain benchmarks. For example, check out the performance of `462.libquantum` in this result! SPEC permits the use of autoparallelization as long as it's clearly indicated.

Unfortunately, this compiler feature is so widely enabled, I couldn't simply exclude all such results. If I had done so, I would be left with zero results for Intel's Core i3, i5 and i7 processor families.

The compromise I chose was to identify the top six benchmarks which seem to benefit from automatic parallelization, disqualify those benchmarks from the test suite, and take the geometric mean of the remaining ones. This approach assumes that automatic parallelization does not work on every benchmark. For the list of disqualified benchmarks, and the algorithm which identifies them, check the GitHub files.

In the end, you'll find that even if you leave the disqualified benchmarks in the results, it doesn't significantly change the conclusions in this post. It shifts most of the CPU2006 results upwards – up to 25% – which simultaneously shifts the conversion ratios from CPU95 and CPU2000 upwards, keeping everything roughly in line.

In the future, it would be interesting for licensees to submit more results without automatic parallelization. That would help us more easily observe the performance trend of single CPU cores.

[Update: HenkPoley has posted an update of these graphs for February 2014.]

« A C++ Profiling Module for Multithreaded APIs Roll Your Own Lightweight Mutex »

Comments (35)

Commenting Disabled
Further commenting on this page has been disabled by the blog admin.

Samuel Williams · *628 weeks ago*

Fantastic graphs - great job!

Reply

Aaron Davies · *628 weeks ago*

Nice graphs. Just out of curiosity, what exactly are they normalized *to*? What's at 1.0 on each graph?

Reply **2 replies** · *active 544 weeks ago*

Jeff Preshing · *628 weeks ago*

They're adjusted CPU2006 results. (I excluded six benchmarks from the calculation.) You can look at this number as a ballpark figure for single-threaded performance relative to a Sun Ultra Enterprise 2.

Reply

Aaron Davies · *628 weeks ago*

Ah, thanks much! (I'm not terribly familiar with SPEC's methodology.)

Reply

Eas · *628 weeks ago*

Very interesting.

Seems like an argument can be made for leaving the autoparallelization influenced results uncorrected. These benchmarks have always tested CPU+compiler. If the compiler can eek more performance out of single threaded code by spreading it over multiple cores, why shouldn't that count?

I'm curious what the trend line looked like bck into the early 90s. At just about the point your grap starts, Intel lost a significant part of the market for high-performance programmable digital logic to the 3D chip vendors. It is also about the time we started seeing SIMD instructions in mainstream CPUs. I wonder what impact those things had on investment in single threaded performance.

Reply **1 reply** · *active 544 weeks ago*

Jeff Preshing · *628 weeks ago*

I'm sure SPEC has legitimate reasons for allowing autoparallelization in their results. For example, if a customer has a huge single-threaded codebase that they just want to run faster by any means, CPU2006 can help them choose a system configuration. That's useful.

But I think there are several reasons to focus on purely single-threaded processes only:

Complex software will become more and more constrained by Amdahl's Law, and it's useful to quantify to what extent we are, and will be, limited by this law.
It gives us a better idea how the CPU microarchitecture itself has evolved.
In a true single-threaded process, we know that the other cores are available to do work. You can launch N copies of the process and expect up to N times the throughput (assuming no shared bottlenecks). But when an application has been auto-parallelized, the same can no longer be said, so the comparison is not completely fair, in my opinion.

Reply

James R · *624 weeks ago*

Truly excellent work - just what I was looking for. Now what I'm curious about is what the industry predicts will be the trend over the next several years. Now that a high-end desktop already has 6 cores, and given that most software isn't written to take advantage of that, what is the incentive to the consumer (the all-important economic driver that you mention) to buy ever-faster CPUs if "ever-faster" means "Well, a little faster, but mostly with more cores that, in many situations, won't do much for you."

In other words, does the industry see the trend as being to dozens of cores and beyond, with only minor improvement in clock speed? Or will they have to figure out how to continue to improve single-threaded performance at a brisk pace to satisfy consumers? I find it a little disheartening that the prevailing trend seems to say "If you need single-threaded performance (and many of us do), Moore's Law is over for you."

Reply

sathyanarayanan · *613 weeks ago*

Great Study !!

I Appreciate you

Reply

Jim · *603 weeks ago*

You went to the trouble of plotting thousands of data points in Excel, but you eyeballed the trendlines?!

Could you post the XLS files so we can do the regression analysis?

Reply **1 reply** · *active 544 weeks ago*

Jeff Preshing · *603 weeks ago*

I didn't use Excel in any way, but you could export the data to Excel yourself by modifying the scripts if you like.

Reply

Ed Austin · *594 weeks ago*

Interesting.

I find using a SPARC T1 (8 x 1GHZ Core, 32 Thread) based system with php saturates a single core, leaving 7 cores untouched except for negligible OS overhead.

In fact the single-threaded'ness of php seems to be conveniently ignored by most of the world but it is highly irritating when you are writing batch mode shell scripts. Granted webservers can spawn separate processes that then exploits multi-core (T1 is ideal for this) but I needed a fast single-threaded CPU to run scripts.

What did I do?
Went out and purchased a dirt cheap Pentium 4 3.8GHZ (harder to find at that speed than you might imagine) and now my scripts trundle along at least with some speed.

Faster than a DC/QC for single threaded php batch jobs.

Reply **2 replies** · *active 544 weeks ago*

fhnjfnvc · *592 weeks ago*

Pentium 4? You would get more single-thread performace per $ with Core2Duo or Core i3/5/7.

Reply

azamat · *580 weeks ago*

I found it odd that you went for a pentium 4, especially since your comment is dated oct 2012. and by purchased I hope you meant scavenged from a dumpster.

Reply

A.Antonio Balaguer · *570 weeks ago*

Thanks interesting. The key assumption in your article is the line you draw. Different "eye balling" could result in 30%-40% growth, not showing a stall. Probably the use of some sort of regression can give a line that is more objectively justified.
Any chance to get the graphs only for server chips and then discriminated charts per processors family (Xeon E3, Xeon E5, etc).

Reply

moneyclip · *542 weeks ago*

Nice graphs and information, thanks a lot!

Reply

Peter · *526 weeks ago*

January 2014 - yes, very slow development. I was wondering if I should upgrade my old Athlon64. Usually I upgrade whenever the budget CPU is five times faster then the older one. I had no doubts when replacing my first 386SX @16MHz with a 486 @66MHz some 5 years later. No doubt also when replacing that with an Athlon system a few years later.

But after that, wow, I've just been upgrading my old motherboard with eBay second-hand cpu's. What I find is that my 10 year-old motherboard (socket 939) with an Athlon X2 3800+, 4GB of ram and a 2013 pcie graphics card, is still more or less equivalent with the budget systems out there, even outperforming some of them! Never happened before.Each 5 five years I had to change the system or I would be literally unable to run current software. Now? My only fear is that the hardware will give in. But there's plenty of second-hand material out there.
Your analysis clarified my gut feeling. Thanks!

I think, other reasons for the flop are the fact that today's computer are sold to the masses that don't really understand or care for real efficiency, they just pay for whatever looks stylish, besides, even the slowest PC runs almost everything and games are more dependent on the graphics card (which did evolve). Also, the focus on the mobile market and the lack of competition from AMD helps the stagnation. I'm hopping the RISC architecture will finally bring some muscle into the fight with Intel. I hope that in 5 years time I will have a reason to upgrade!

Reply

Jamie · *520 weeks ago*

Just to say, thank you for your great scripts; they are exactly what I was looking for to collate SPEC data.

I do have a couple of comments:
- I don't think the approach you have taken to exclude a top set of benchmarks that benefit from parallelism does not characterise sequential/single-thread performance. Based on what has happened with clock speeds and microarchitecture since 2005, I would not expect anywhere near 21% year-on-year growth in sequential performance. If you instead plot the SpecINT score divided by the number of enabled cores, you get quite a different picture.
- There is a very steep increase in performance for Intel devices around 2006 and this becomes more pronounced when you plot /number of cores. I think these may be errors in the data set, perhaps with CPUs recoded as single core when they are not.

Reply **1 reply** · *active 518 weeks ago*

Jamie · *518 weeks ago*

In response to the last point in my last comment, I think this must in fact be due to the scale factors calculated for the non-2006 data sets not being quite right. I don't think it is coincidence that it lies exactly at 2006 although it is still strange that it only affects Intel processors. I wonder then if there might be a better way of calculating the scaling factors?

Reply

Ananya M · *517 weeks ago*

Before reading your article, I was under the impression that single-threaded performance is going down with a negative slope. You have now given me a valuable counter-insight. Thank you! And thank you for explaining how you have adjusted the graphs.

Reply

henry · *501 weeks ago*

Instead of "eyeballing" it, why didn't you just run a regression?

Reply **1 reply** · *active 469 weeks ago*

dcarter1979 · *469 weeks ago*

You can download the scripts at github and rerun it, put it excel and run it yourself if you would like. Please post the results back here.

Reply

pac_71 · *449 weeks ago*

Only just stumbled across your study. I did a similar thing back in 2008 and recent stagnation of mobile CPU performance has led me to revisiting my study which i have written about here .... http://justinfromthesandbox.blogspot.com.au/2015/...

Reply

henkpoley · *449 weeks ago*

Fetched the data another time. Data until June 2015: http://imgur.com/a/r3wbh

Fixed an off-by-one bug in the rendering, should use math.ceil() instead of round() for maxLogScore.

Reply **1 reply** · *active 433 weeks ago*

normus · *433 weeks ago*

About your comment on the lone i7-5960X in the update, that data-point shows the processor is overclocked (there is at least one more on the data as of November).

Reply

Evan · *432 weeks ago*

In the python code there is a report.txt file needed to run, would you be able to provide that? Also, would you happen to have the data points from rendering the graph? Thanks, The graph is Awesome!!!

Reply **1 reply** · *active 432 weeks ago*

preshing · *432 weeks ago*

Thanks. I don't have them handy anymore, but you can generate them by following the instructions in the README.

Reply

Evan · *430 weeks ago*

I was able to run the fetch-pages.py, however, when I run the analyze-pages.py, I get an error saying windows cannot find the specified path, for the scraped CPU 95. I'm not sure why theres an error.

Reply

@LaurentIIHE · *414 weeks ago*

Thanks for your work.

Just 2 questions : does the vectorization features of the core are activated in the bench? And do you think (have the information) that without SSE, AVX the curve will be flat?

Reply

baptistezh · *394 weeks ago*

thanks for your excellent work really

I've used your script to update the tendance to 2016
here's link https://img3.doubanio.com/view/status/raw/public/... https://img3.doubanio.com/view/status/raw/public/...

however, how can I draw the red average line like you did in this article?
Thank you very much

Reply **1 reply** · *active 375 weeks ago*

SuperNerd · *375 weeks ago*

I get a 403 error when trying to access your pictures. Regarding the red trendline, this might do it: http://stackoverflow.com/questions/26447191/how-t... Scikit learn will probably have more advanced fits.

Reply

KGT UserCast YT · *377 weeks ago*

I wonder where the AMD FX 9590 would be at. Or a i7 6700K. But it seems like clock speed increases have stagnated. I dont see any Intel CPUs over 4.5 GHz. Perhaps the only way for CPUs to improve single thread performance is to aim for better IPC, or cache/memory latency. But IPC gains are limited too, particularly past Kaby Lake/Zen/Zen+. Then we can only rely on smaller lithography for that little bit of efficiency and boost clock rates ever so slightly (P=C V^2 F) obeying a cubic power increase. After that, I think we are doomed to no more CPU single thread advances, and either to proceed on to manycore architectures of GPUs or more on to Indium Gallium Arsenide (InGaAs), Graphene, Optical Integrated Circuits, or Quantum computers.
I bet my money on InGaAs. 10 GHz approaching!

Reply

Todd Stewart · *372 weeks ago*

Any update on these numbers would be amazing!

Reply **2 replies** · *active 304 weeks ago*

preshing · *372 weeks ago*

Here's one up to the end of 2015: https://twitter.com/HenkPoley/status/675007874448...

Reply

henkpoley · *304 weeks ago*

One from today, 25th April 2018: https://imgur.com/a/frbo6AX

Reply

Dave B · *341 weeks ago*

I do not believe that your methodology is actually measuring single-threaded CPU performance. It is measuring overall system performance (including the compiler) as reported by people who have a vested interest in making the performance against SPEC look as good as possible.

To accurately measure CPU only you would need to compile the code and run the exact same executable on the spectrum of processors

from the various vendors and architectures across the years of interest. The hardware (not just CPU) vendors, in the absence of clock rate increases since roughly 2001, have added hardware specifically to make their SPEC numbers look better. Those improvements may or may not help in real workloads, but they make the benchmarks better. Similarly, they will make compiler improvements that target specific things in the various benchmarks to improve the marketing results without necessarily having any effect on real-world performance.

I do not mean to in any way diminsh the value of your efforts and the applicability and usefulness of the results you are producing, but I believe that you and the guys from nVidia and AMD are measuring different things. I suspect that if you did the test I recommend here, you'd find that the AMD guy is probably being a little optimistic in his results as well (he does, after all, work for one of the companies who needs to show improvements here).

Reply
