CHALMERS

An In-Depth Look at Computer Performance Growth

Magnus Ekman

Fredrik Warg

Jim Nilsson

Technical Report 2004-9

CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Computer Engineering
Göteborg 2004

An In-Depth Look at Computer Performance Growth

Magnus Ekman
Department of Computer Engineering
Chalmers University of Technology
Email: mekman@ce.chalmers.se

Fredrik Warg
Department of Computer Engineering
Chalmers University of Technology
Email: warg@ce.chalmers.se

Jim Nilsson
Department of Computer Engineering
Chalmers University of Technology
Email: j@ce.chalmers.se

***Abstract*— It is a common belief that computer performance growth is over 50% annually, or that performance doubles every 18-20 months. By analyzing publicly available results from the SPEC integer (CINT) benchmark suites, we conclude that this was true between 1985 and 1996 – the early years of the RISC paradigm.**

During the last 7.5 years (1996-2004), however, performance growth has slowed down to 41%, with signs of a continuing decline. Meanwhile, clock frequency has improved with about 29% annually. The improvement in clock frequency was enabled both by an annual device speed scaling of 20% as well as by longer pipelines with a lower gate-depth in each stage. This paper takes a fresh look at – and tries to remove the confusion about – performance scaling that exists in the computer architecture community.

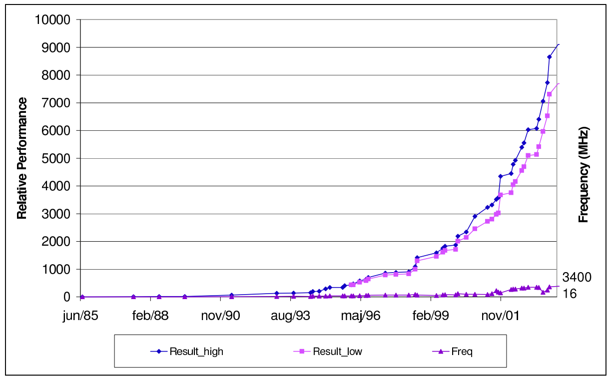

Fig. 1. Computer performance 1985-2004, normalized to SPEC CINT89.

I. Introduction

Computer performance has grown exponentially for decades, driven by a combination of improvements in implementation technology, architectural innovations, and compiler optimizations. Without doubt, the performance growth rate has been spectacular. During the years, there have been a number of attempts at quantifying this performance growth. An old number of processor performance that has been associated with Moore's Law states that performance doubles every 18 months [8]. Roberts [7] reports a doubling every 21 months over a 41 year period.

Another often cited number is 58% annual performance growth given by Hennessy and Patterson in [5] and repeated in [6]. Analyzing the performance numbers given in [6], however, reveals that performance growth in more recent years is lower. For instance, between 1992 and 2000 performance only seems to grow 35% annually. This number piqued our interest and prompted this investigation which spans the years 1985-2004, starting with the transition to RISC architectures. We concentrate on single-thread integer performance measured with the well-known SPEC CPU integer benchmarks (CINT).

Our numbers, plotted in Figure I, are not as bleak as those found in [6], but still far from the often-cited 50-60% per year, or doubling every 18-20 months. During a period of relatively stable growth between late 1996 and early 2004 (7.5 years), performance grew about 41% annually. This performance increase has been enabled by organizing the new and faster resources from technology scaling in an intelligent way and to some extent by better compilers. The studied architectures have had an annual growth in frequency of 29%, due to both deeper pipelines with shorter stages and device speed scaling. We have isolated the effects of device scaling and found that it alone would only have yielded a 20% freqency increase. Finally, the average number of instructions per cycle (IPC) remains close to one.

The motivation behind this work is to establish some facts about how performance has scaled and where we are today. The need for this is obvious if the proceedings from ISCA 2003 is studied. Seven papers presented CINT2000 IPC – baseline numbers ranged from 1.5 to 2.3. It almost seems like people are afraid of publishing lower numbers. This might lead researchers to use unrealistic parameters and assumptions in their simulations. For instance, some papers assume memory latencies of 50 cycles, which was far from true at the time the paper was published. As a result, the positive effect of micro-architectural mechanisms that do not aim at reducing or tolerating cache misses may easliy be over-estimated.

II. Methodology

How to measure computer performance and choose fair benchmarks is a contended issue, and beyond the scope of this paper. However, the SPEC CPU benchmarks have long been the predominantly used measure of performance among computer architects. In addition, the SPEC CPU integer benchmark, CINT (formerly known as SPECint) is the basis for the well-known figure 58% annual performance growth [5], and finally, measurements on top-of-the-line machines are readily available. Therefore, in this paper we equate computer performance with CINT performance. We note, however, that the performance development is likely to look different depending on the workload used for comparison. All data used, unless otherwise noted, are from publicly available databases[1].

The SPEC benchmark suites are updated every few years in order to adjust the workloads for modern machines. Measurements on top-of-the-line machines are typically only available for the latest benchmark. This is for good reasons, since old benchmarks are likely to have footprints small enough to fit in on-chip caches on modern machines and are not representative of contemporary workloads, while new benchmarks have too large workloads to run on old machines. Unfortunately, this makes performance comparisons difficult. We have chosen to treat each benchmark suite as representative of its time, stressing both CPU and the memory system. Like in [5], we attempt to normalize results from different benchmarks to the same relative performance scale. Results are dated to the month they were reported, which is typically close to when all hardware and software used was available. There are, however, some results which were clearly from machines much older than the test date, and where hardware availability dates were used instead. All data that was used can be found in Table II.

A. Normalizing Results from Different Bechmark Suites

In order to normalize the results from one benchmark suite to another, we need to find results for both benchmarks from the same (base) machine. We then normalize e.g. a CINT2000 result to a CINT95 result simply by multiplying it with the ratio between the CINT95 and CINT2000 results on the base machine. However, this ratio depends on the machine used for normalization. In order to strengthen our conclusions, we have chosen to use two different machines for each benchmark transition. This leaves us with four performance numbers for the most recent machines, since there are two benchmark transitions where normalization is needed. We report the highest and lowest performance numbers of these four (listed in columns High and Low in Table II). Since the difference in annual growth between the high and low performance numbers is very small, we feel confindent that our results are not significantly biased by the normalization.

B. Beware of Benchmark Transitions

When compiling peak performance numbers to see performance development, it is important to make sure the results really are from top-of-the-line machines. This is not always the case if you simply pick the highest currently available performance number. For instance, picking top reported CINT95 results for Alpha processors in late 1995, and early 1996 would encourage you to believe the performance growth was about 400% in a few months. This is not the case; the first reported CINT95 results were actually for a machine released in 1992, but since CINT95 was introduced in 1995, the test date was from that year. The same is true when the IBM POWER3 is the CINT2000 performance leader for a brief period. In both cases these machines seemed to be performance leaders simply because there were no other reported results for the new bechmark.

Another caveat is that as we normalize different benchmarks we also postulate that the relative performance is the same for both benchmarks on that machine. The percentual performance growth before and after the switch is correct, but you cannot easily compare other properties across this boundary. Especially, since the working sets of the newer benchmark is scaled to better match contemporary workloads, the average IPC is likely to be lower. This means if we have CINT95 IPC on a machine it is not true that the relative performance advantage of a machine measured with CINT2000 can be derived from frequency and IPC increase; the effects of application scaling is also included in the performance gain.

III. Interpreting the Numbers

Figure 1 plots relative performance, normalized to the CINT89 performance scale, as well as clock frequency from 1985 to 2004. There are three curves in the figure. The two upper curves describe the performance depending on the normalization used in the benchmark transitions. The highest curve peaks at about 9000 and represents the most optimistic performance growth, while the middle curve represents the pessimistic growth and peaks 16% lower. The clock frequency peaks at 3.4 GHz. In 1985 the relative performance was about 1.8 while frequency was 16 MHz, leading to a total increase in performance of more than 5000x, while clock frequency has scaled over 200x. This translates into an annual growth in performance of 56% or 58% (depending on the normalization), and 33% in clock frequency. Since the chosen normalization does not affect the result much, we have chosen the one yielding a 58% annual growth in the rest of this paper.

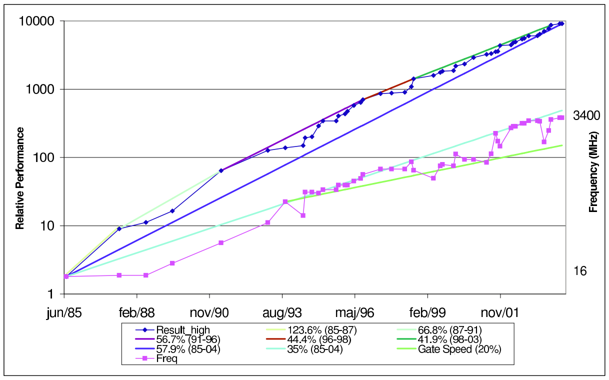

Fig. 2. Computer performance 1985-2004, logarithmic scale.

With exponential growth scenarios it is often more informative to consider a logarithmic scale. This is shown in Figure 2. There are several curves plotted in the figure. Result_high plots the performance of the currently best reported results during the entire time frame. It starts at 1.8 and continues in a shaky walk up to 9102. The following five curves connects the peaks of the performance curve and forms the convex hull.

[1] http://www.spec.org, http://performance.netlib.org, and Usenet archives.

These curves can be looked upon as the general performance trend. The slope shows that performance growth has been decreasing during the entire period. This can be explained by the remarkable improvements with the introduction of RISC (123.6% annual growth from 1985 to 1987). Performace growth then gradually slows down and in 2004 meets the straight line 57.9%, which shows average annual growth for the period 1985-2004.

The second straight line in the graph shows 35% annual growth which is the rate with which transistor density has increased. In [6] this is quoted as technology scaling, but it is important to understand that various properties of the technology scale in different ways. While transistor density scales with 35% annually, die sizes also increase with about 10-20% each year [6]. These two effects lead to about 55% increase in transistor count per chip.

The straight line which spans from 1993 to 2004 shows how gate speed has increased since the introduction of the Alpha 21064, and is normalized to the frequency of the first Alpha. This line, with an annual growth of 20%, tracks how frequency would have scaled if the architecture would not have been changed. This growth was calculated in various ways. The first way was by combining numbers from [4], which shows how gate delay scales with feature size, with the actual processes used for a number of Alpha and Pentium 4 chips. Another way was to look at the actual clock frequency increase for Alpha. However, in addition to using new processes for the newer generations, the architecture was also changed so cycle time decreased with two gate delays (from 16 through 14 to 12) for each new generation [3]. Then the frequency scaling was calculated with the effect of the architectural change eliminated. Both these calculations resulted in an annual growth of about 20%.

It is worth noticing that since the introduction of Alpha, frequency has scaled more like transistor density than like gate speed. As mentioned above, Alpha managed to do that by decreasing the gate depth between each pipeline stage. Pentium 4 further increased the frequency by having a longer pipeline.

Looking at the convex hull in the graph there are no knees in the last years, with the exception of a small knee in 1998 where performance growth decreases from 44.4% to 41.9% annually. In our further analysis we consider late 1996 to 2004 as a single 7.5 year period with a fairly stable annual growth of around 41%. This is a clear change compared with the previous period (1991-1996) with a growth of 57%.

A. Performance Growth The Last 7.5 Years

During the first part of this period, Alpha was the dominating processor. In 2001, Pentium 4 took the lead and has been dominating ever since. Even though other processors sporadically show up as performance leaders, these two clearly dominate. Figure 3 shows how frequency (upper curve) and the ratio between performance and frequency (lower curve) have evolved during these years. The ratio between performance and frequency is a measure on how well the processor can exploit ILP but also includes compiler improvements and new instructions. It is also important to understand that the two processor families have fundamentally different instruction set architectures (ISAs). We see that the frequency is steadily increasing within a processor family, while ILP increases in steps with the introduction of a new model.

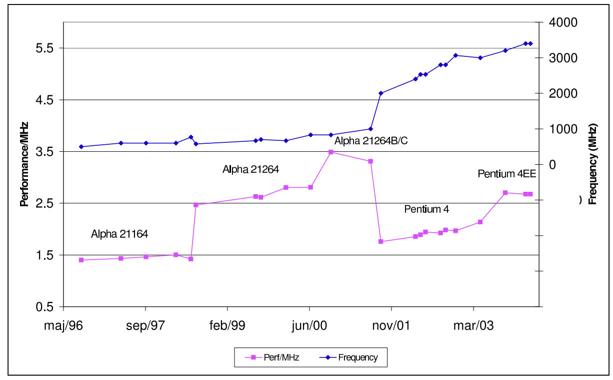

Fig. 3. Performance per MHz 1993-2004.

A very interesting point is the transition between Alpha and Pentium 4. The frequency doubles from 1 GHz to 2 GHz, much due to the Pentium 4's much longer pipeline, while ILP decreases by a factor of two. This is not the first time in history that this behavior is observed. In fact, when Alpha was introduced, it was clocked at a very high frequency compared to other processors at the time, but performance was not proportionally higher, leading to a similar fall in performance/MHz. Performance/MHz is not readily translated to other commonly used metrics. However, according to [2][2] the average number of instructions per cycle (IPC) for Alpha 21264 is 1.05 for CINT95. This corresponds to about 2.6 in Figure 3. According to the same figure, Pentium 4 has about the same performance/MHz today. With the reasoning about IPC in Section II-B in mind, the IPC of a Pentium 4 should be lower today than for the Alpha. In Table I we measure IPC when running CINT95 and CINT2000 on the same machine, a 600 MHz SunBlade1000 from year 2000. The CINT2000 IPC is about 14% lower. We find it likely that the result for other architectures would be similar. Thus, it seems like computer architects are still struggling to achive an IPC over one. High frequency is a concious design choice which has so far yielded the best performance, but pushes IPC down; however, it seems this may change soon due to power/heat problems.

B. Discrepancies From Other Growth Figures

As stated in Section I, the graph in [6] shows a 35% annual performance growth between 1992 and 2000 (on p. 3, performance grows from about 140 to 1550) while our numbers show about 50% for the same time period. We see two possible reasons for this large discrepancy. It might have to do with how the results are dated. We see a rapid improvement with the introduction of the 99 MHz PA-RISC 7100 which is

[2] An errata for this paper was later published but does not affect this number.

dated Feb-93 according to Table 1, while in [6], it seems like that this machine is introduced in 1992. But this does not explain everything, the normalized results for Pentium III in year 2000 is significantly lower in [6] than our numbers. We suspect it has to do with how results are normalized.

In [1], the growth number 50%-60% is not explicitly backed up with references, but performance graphs shown in the paper can be seen as implicitly backing up the statement. When considering the graphs into more detail, however, it is seen that they include erroneous points as described in Section II-B. Other growth numbers cited in Section I are either out-dated and/or use other benchmarks and time-frames.

IV. Conclusions & Future Work

During the past 7.5 years we have seen an annual performance growth of about 41%, and clock frequency growth of 29%. The increase in clock frequency has been enabled by both technology scaling and by new architectures. This paper has tried to remove the confusion surrounding performance scaling and shown that there is a clear trade-off between ILP and frequency. Both of them are effective ways of increasing performance, but so far we have not seen any architecture which has been able to sustain a high IPC (over one), with a top-of-the-line frequency, in spite of the high numbers reported in simulation studies.

There are still lingering questions not revealed by SPEC CINT figures alone. For instance, how have architectural improvements and compiler technology affected performance growth, i.e. where would we be with technology scaling alone? We hope to answer some of these questions in future work, giving a clearer picture of the history and future of computer architecture and performance.

Acknowledgments

We thank Lars Svensson, Martin Thuresson, and Charlotta Bååth for valuable comments and discussions.

References

[1] V. Agarval and M.S. Hrishikesh and S.W. Keckler and D. Burger, *Clock Rate versus IPC: The End of the Road for Conventional Microarchitectures*, In Proceedings of the 27th Annual International Symposium on Computer Architecture (ISCA 2000), ACM Press, June 2000.

[2] R. Desikan and D. Burger and S.W. Keckler, *Measuring Experimental Error in Microprocessor Simulation*, In Proceedings of the 28th Annual International Symposium on Computer Architecture (ISCA 2001), ACM Press, July 2001.

[3] Paul E. Gronowski, William J. Bowhill, Ronald P. Preston, Michael K. Gowan and Randy L. Allmon, *High-Performance Microprocessor Design*, Journal of Solid-State Circuits, vol. 33, 676-686, 1998.

[4] Carver A. Mead, *Scaling of MOS Technology to Submicrometer Feature Sizes*, Journal of VLSI Signal Processing, 8, 9-25, 1996.

[5] J.L. Hennessy and D.A. Patterson, *Computer Architecture: A Quantitative Approach*, 2nd ed. Morgan Kaufmann Publishers, 1996.

[6] J.L. Hennessy and D.A. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd ed. Morgan Kaufmann Publishers, 2003.

[7] L. Roberts, *Beyond Moore's Law: Internet Growth Trends*, Internet Watch, Jan 2000.

[8] I. Tuimi, *The Lives and Death of Moore's Law*, First Monday, Oct 11, 2002.

[3]Crafty is not included due to problems running it.

TABLE I

IPC for CINT2000 and CINT95 Applications on SunBlade1000.

CINT2000[3] avg: 0.69 median: 0.68									
bz2	gzp	gap	gcc	mcf	two	vor	vpr	par	prl
.85	.92	.65	.67	.18	.57	.79	.62	.69	.82
CINT95 avg: 0.80 median: 0.78									
	go	ijp	com	li	gcc	prl	m88	vor	
	.82	1.01	.84	.63	.71	.72	.91	.74	

TABLE II

Summary of SPEC CPU Benchmark Results.

Processor	**Date**	**MHz**	**High**	**Low**	**CINT[1]**
Pentium 4 EE	Mar-04	3400	9102	7691	1705 (00)
Pentium 4 EE	Feb-04	3400	9096	7686	1704 (00)
Pentium 4 EE	Oct-03	3200	8648	7307	1620 (00)
Athlon FX-51	Sep-03	2200	7725	6527	1447 (00)
Itanium 2	Jul-03	1500	7057	5963	1322 (00)
Pentium 4	May-03	3000	6406	5413	1200 (00)
Intel Xeon	Apr-03	3066	6073	5133	1138 (00)
Pentium 4	Dec-02	3066	6032	5097	1130 (00)
Pentium 4	Oct-02	2800	5552	4691	1040 (00)
Pentium 4	Sep-02	2800	5392	4556	1010 (00)
Pentium 4	Jun-02	2533	4922	4159	922 (00)
Pentium 4	May-02	2533	4783	4042	896 (00)
Pentium 4	Apr-02	2400	4447	3757	833 (00)
POWER4	Nov-01	1300	4345	3672	814 (00)
AthlonXP 1800+	Oct-01	1533	3582	3027	671 (00)
Pentium 4	Sep-01	2000	3518	2973	659 (00)
Alpha 21264C	Jul-01	1000	3315	2801	621 (00)
HP PA-8700	May-01	750	3224	2724	604 (00)
Alpha 21264B	Nov-00	833	2904	2454	544 (00)
Alpha 21264	Jul-00	833	2338	2143	50.0[3] (95)
Alpha 21264	Jul-00	833	2338	2143	533[3] (00)
Pentium III	Mar-00	1000	2189	2005	46.8[3] (95)
Pentium III	Mar-00	1000	2189	2005	410[3] (00)
Alpha 21264A	Feb-00	667	1871	1714	40.0 (95)
Alpha 21264A	Sep-99	700	1829	1675	39.1 (95)
Alpha 21264A	Aug-99	667	1754	1607	37.5 (95)
PA-RISC 8500	May-99	440	1590	1457	34.0 (95)
Alpha 21264	Aug-98	575	1417	1298	30.3 (95)
Alpha 21164	Jul-98	767	1090	998.4	23.3 (95)
Alpha 21164	Apr-98	600	902.6	827.0	19.3 (95)
Alpha 21164	Oct-97	600	879.2	805.6	18.8 (95)
Alpha 21164	May-97	600	860.5	788.5	18.4 (95)
Alpha 21164	Sep-96	500	701.5	642.8	15.0 (95)
Alpha 21164	Aug-96	437	636.0	582.8	13.6 (95)
Alpha 21164	May-96	400	575.2	527.1	12.3 (95)
Alpha 21164	Feb-96	350	472.4	432.8	10.1[3] (95)
Alpha 21164	Jan-96	350	432.8	432.8	432.8[3] (92)
Alpha 21164	Oct-95	350	405.9		405.9 (92)
Alpha 21164	Mar-95	300	341.4		341.4[3] (92)
Alpha 21164	Sep-95	300	341.4		7.3[3,4] (95)
Alpha 21164	Jan-95	266	288.6		288.6 (92)
Alpha 21064A	Oct-94	275	200.6		200.6 (92)
Alpha 21064A	Jul-94	275	193.8		193.8 (92)
PA-RISC 7150	Jun-94	125	149.4		149.4 (92)
Alpha 21064	Oct-93	200	138.4		138.4 (92)
PA-RISC 7100	Feb-93[2]	99	126.8		126.8 (92)
PA-RISC 7000	May-91[2]	50	64		64.0 (92)
MIPS R3000	Jul-89[2]	25	16.4		16.4 (89)
MIPS R2000	Jul-88[2]	16.67	11.2		11.2 (89)
SPARC (SUN4)	Jul-87[2]	16.67	9		9 (89)
68020 (SUN3)	Jul-85	16	1.8		1.8[5]

[1] CINT version indicated in parenthesis.
[2] Machine introduction date, no earlier scores exist.
[3] Used to normalize the CINT92 to '95 and '95 to '2000.
[4] Result is not performance leader; needed only for normalization.
[5] Perf. estimated from MIPS-rating difference between SUN3 and SUN4.