DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

ONDERZOEKSRAPPORT NR 9631

GROSCH'S LAW: A STATISTICAL ILLUSION?

by

Herlinde Leemans

Guido Dedene

Katholieke Universiteit Leuven

Naamsestraat 69, B-3000 Leuven

ONDERZOEKSRAPPORT NR 9631

GROSCH'S LAW: A STATISTICAL ILLUSION?

by

Herlinde Leemans

Guido Dedene

D/1996/2376/31

GROSCH's LAW:
A STATISTICAL ILLUSION ?

Herlinde LEEMANS
Prof. Dr. Guido DEDENE
Katholieke Universiteit Leuven
Faculty of Economics and Applied Economics
L.I.R.I.S.
The Leuven Institute for Research on Information Systems
Naamsestraat, 69
B-3000 Leuven
Belgium
Fax 00 32 16 32 67 32

Abstract

In this paper a central law on economies of scale in computer hardware pricing, Grosch's law, is discussed. The history and various validation efforts are examined in detail. It is shown how the last set of validations during the eighties may be interpreted as a statistical misinterpretation, although this effect may have been present in all validation attempts, including the earliest ones. Simulation experiments reveal that constant returns to scale in combination with decreasing computer prices may give the illusion of Grosch's law when performing regression models against computer prices over many years. The paper also shows how the appropriate definition of computer capacity, and in particular Kleinrock's power definition, plays a central role in economies of scale for computer prices.

Keywords

Economies of scale - Computer pricing - Computer capacity - Computer performance - Computer Business.

0. Introduction: a motivating discounted cash flow exercise

Ever since computers exist, the question on economies of scale in computer price is present. A central result that is still persistent in computer literature today [GILDER 1994] is Grosch's law, formulated as early as in the fifties. The law can be formulated as follows :

$$c = aw^{0.5}$$

where

c	represents the hardware costs of a computer system
w	is a measure of the capacity of the computer system
a	is a proportionality constant

This law on the cost of computing survived for several decades. It favorizes heavily the acquisition of large machines, since the cost per unit of capacity goes down as the capacity increases. Nevertheless a very straightforward discounted cash flow exercise reveals that this law is somewhat dubious.

Example

Compare the following computer system acquisition scenarios, for a system that requires a computer capacity of $2w$ at the end of two time periods of length T during which the interest rate is r, assuming that the demand increases linearly and computer prices remain constant over the two time periods :

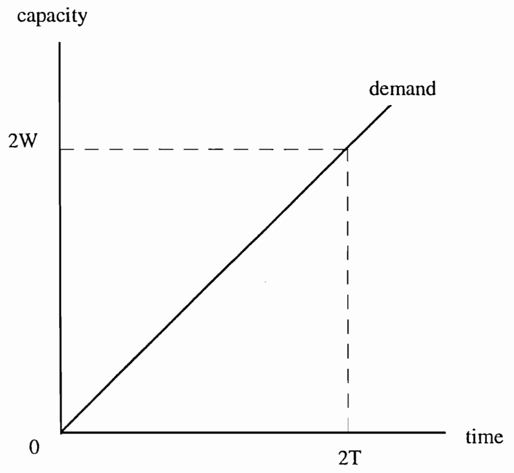
Scenario A

Buy a system of capacity $2w$ at time *0*

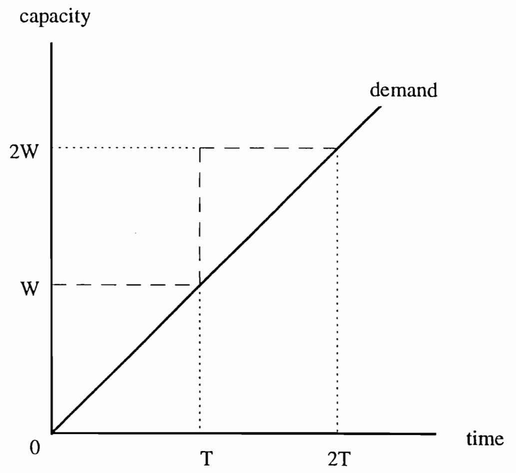
Scenario B

Buy a system of capacity w at time *0* and buy another system of capacity w at time T

Scenario A



Scenario B



Following the principles of capacity planning, which consider excess capacity as a waste, one might expect scenario B to be cheaper than scenario A. If Grosch's law would be true, surprisingly this turns out not to be correct. In fact, the cost for the scenarios is the following :

$$C_A = a(2w)^{0.5}$$

$$C_B = aw^{0.5} + (aw^{0.5})/(1+r)$$

Comparing the two costs reveals that $C_B < C_A$ if $r > 2^{0.5} = 141\%$, which is only the case for long time periods T or high inflations (over 100 % during one time period).

Hence the consequences of Grosch's law are less trivial than might be expected at first. However, at regular points in time, validations of this law have been published. During the eighties there was even a very interesting and animated series of papers discussing Grosch's law type of effects in computer prices [EIN-DOR 1985a, EIN-DOR 1985b, JONES 1985, KANG, MILLER & PICK 1986, KLEINROCK 1987, MENDELSON 1987], there has never been a real explanation for the fact that this law could persist for many decades. This paper is an attempt in this direction. It will be shown how a regression model on computer prices over many years, with constant returns to scale within each year and decreasing computer prices on a yearly basis, may lead to the illusion of a Grosch's law effect.

The paper is organized in the following way. First an overview of the amazingly long history of Grosch's law is given. The main result of the paper is developed in the next section, featuring a simulation experiment against the validations of the eighties on Grosch's law. Next, the various concepts of computer capacity and speed are discussed, in order to position Grosch's law type of effects appropriately. The paper concludes with a discussion of the results, arguing whether Grosch's law ever existed.

1. Grosch's Law: a story of validation and criticism

Grosch's law was published more or less as an economic hypothesis, without fundamental theoretical foundations [GROSCH 1953]. At the time of its formulation it was in no way based on empirical data. Validation of Grosch's law is not so difficult, provided that a uniform allocation of data for the cost and the capacity is used. Indeed, it suffices to take the logarithmic version of the law and perform a regression analysis as follows :

$$\log(c) = \log(a) + b\log(w)$$

The factor b, which is to be estimated with a sufficient degree of determinism (measured by the R-square of the simple linear regression model) should be as close as possible to 0.5, which is Grosch's coefficient.

The first validation exercises have been published in the sixties [KNIGHT 1966, 1968]. Knight published the following Grosch's coefficients :

Period	*Scientific Computing*	*Commercial Computing*
1950 - 1962	0.519	0.459
1962 - 1966	0.322	0.404

At first sight, these studies seem to confirm Grosch's law. However, a crucial observation that formed the start of the research of this paper is the fact that Grosch's law was validated better when data over a longer period are studied : the coefficients resulting from regression over the 1950 to 1962 data are closer to 0.5 than the coefficients for the 1962 to 1966 data. This validation experiment also already showed that the computer market is not homogeneous.

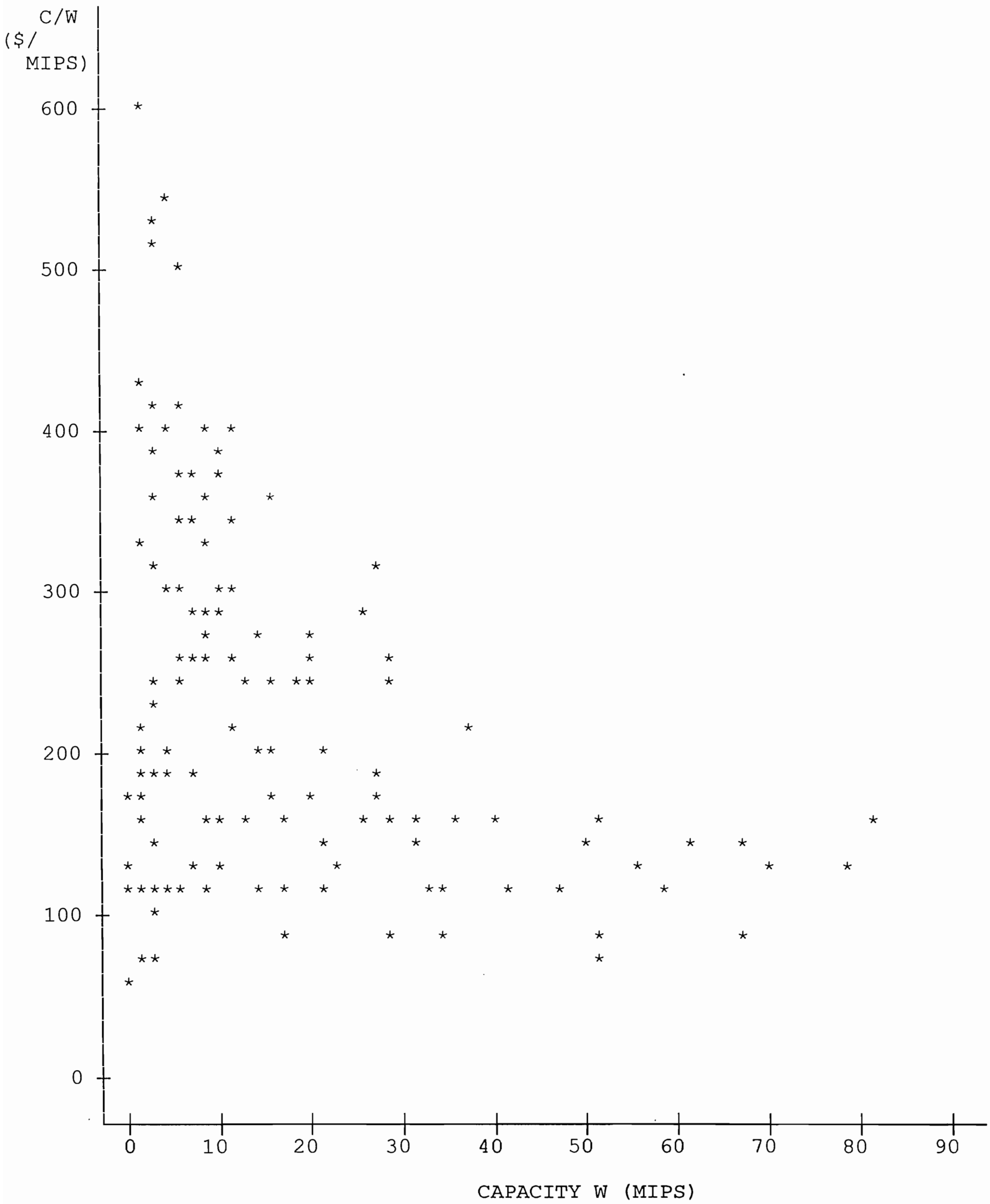
Grosch published a validation of his law in the mid seventies [GROSCH 1975]. At the end of the seventies Cale experimented with alternative attempts to correlate the cost of a computer system to some of its components. The study analyses data over the period 1972 to 1977 and demonstrates a Grosch like coefficient for the correlation between the cost of a computer system and the memory size. Of course, the importance of using the memory size instead of the computer capacity in terms of a processor speed measure can be questioned. Cale's study also showed that analysis on subsets of the data (e.g. General Purpose Systems versus Small Business Systems) gives significantly different results.

A milestone in the history of Grosch's law was the publication of the study by Ein-Dor [EIN-DOR 1985a]. His study, which was based on a set of data published in a Computerworld Hardware Round-up [HENKEL 1981], showed that Grosch's law was no longer valid on this set of data. On the contrary Ein-Dor discovered overall increasing returns to scale on computer costs. However, analogous to the study of Cale, an analysis on five subsets revealed five subsets of computer types for which Grosch's law was valid. Subsequently, an amended version of Grosch law was born, stating that within the smallest computer category that was needed, the largest computer was still favorable. Needless to say that this publication was the start of a very animated series of follow-on papers debating the subject [JONES 1985, EIN-DOR 1985b, KANG, MILLER & PICK 1986, MENDELSON 1987].

A very interesting reaction was the paper by Mendelson [MENDELSON 1987] as it attacks precisely the use of subgroups in the validation of Grosch's law type of effects. Mendelson was using data including 1985 computer prices and found constant returns to scale. Moreover, he showed that the grouping of data into subsets could produce basically any law (on the subgroup data). In other words, any law could be validated provided the appropriate subset grouping is chosen on the data that are used for the validation.

In the paper of Mendelson another striking effect arises : he and Ein-Dor claim to use the same set of data for some validations. Ein-Dor finds increasing returns to scale (1.30 as Grosch's coefficient) while Mendelson reported constant returns to scale (1.03 as Grosch's coefficient). Careful inspection reveals that Ein-Dor had included 11 observations on micro-computers in his data, which were not included in Mendelson's experiments. It is remarkable that about 10% of data can give such a deviation in the regression model.

At the beginning of the nineties some research work was initiated to validate Grosch's law type of effects on the whole set of data available for the eighties, including [HENKEL 1988a & 1988b]. The following is a plot of the test data, showing the average cost per capacity (*c/w*) versus the capacity (*w*).



These observations are shown here because they already visually may give the impression of a Grosch's law type of effect. However, the validation procedure described above leads to the following conclusions on these data:

a) On the overall data there are constant returns to scale (Grosch's coefficient of 0.924), however with a very weak degree of determination (R-square of 0.036).

b) The same effect arises when looking at subsets of data, as published in Computerworld in 1981 (Grosch's coefficient of 0.913), 1985 (Grosch's coefficient of 0.978) and 1987 (Grosch's coefficient of 0.995) with slightly higher degrees of determination.

c) A remarkable effect can be discovered when looking at the maximum prices per unit of computer capacity. A linear regression over the maximum price per unit of capacity over all the data shows a regression coefficient of 0.55, remarkably close to Grosch's coefficient. Moreover these results appear with a strong degree of determination (R-square of 0.693). Of course, this result cannot be interpreted as an effect of economies of scale, since only maximal prices are considered. The only consideration that can be attached to this observation is the fact that for larger capacities the risk of paying more than the average price decreases.

The conclusion on this last validation is double. On one hand there is an ongoing support for constant returns to scale in computer prices, especially when looking at data over a shorter time period. On the other hand there is still the question how the impression of economies of scale could persist for so long. The simulation of the next section will provide some insight.

2. A simulation experiment on Grosch's Law

Stimulated by the above validation exercise, the following scenario will be simulated in this section:

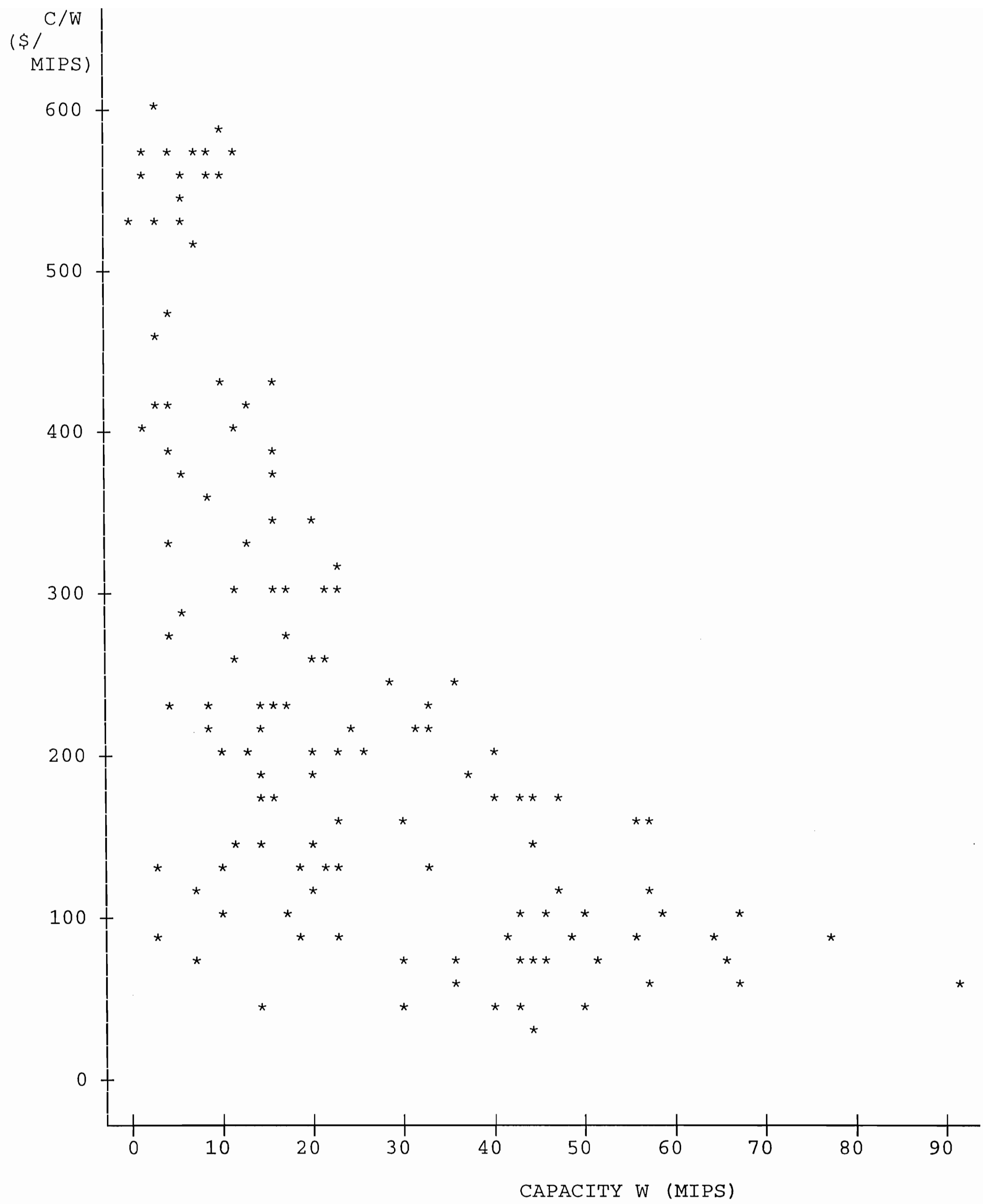
> *Assume that computer prices are constant per unit of capacity during a one year period. Assume at the same time that over several years, computer prices are going down at some rate. During the same period, each year the maximum capacity of computers also increases. Then, what is the result of Grosch's law's validation procedure against these simulated observations ?*

The simulation experiment was designed such that its data set would conform to the validation set used in the previous section. The detailed SAS-code of the simulation can be found in the appendix, but the parameters used are the following:

a) During one year computer capacities are picked randomly from a Gamma distribution up to a specified maximum. The Gamma distribution is used to create a skew distribution that simulates the effect of having more computers with lower capacity in the market. Computer prices are related to the selected capacities by means of a Uniform distribution around a constant price per unit of capacity ("MIPS").

b) During the first year the maximum capacity is limited to 15 MIPS, in accordance with the data from the previous section. Next, for a period of 9 years the average price per unit of capacity goes down by 35%, while the maximum capacity available increases with 25%. (These numbers were also derived from the Computerworld dataset).

The result is the following dataset, which resembles the data from the previous section, as can be seen from visualising the data.



The validation procedure of Grosch's law on this set of data results in a Grosch's coefficient of 0.50 (sic !) with a relatively high degree of determination (R-square equal to 0.51).

It is also instructive to look at the behavior of the maximum price per unit of capacity for comparison with the results of the previous section. For this simulation, the result gives a coefficient of 0.355, with an R-square of 0.73. This result is again close to what was discovered before.

Various alternative simulation experiments give analogue results. Increasing the number of years doesn't change the results, decreasing the number of years makes them less determined. Variations in the Gamma distribution as the years proceed also result in an overall Grosch's coefficient around 0.50.

The result may be interesting, since it shows that in a market with only constant returns to scale, the evolution of prices in time may lead to the impression of economies of scale. These results are complementary to the material of Ein-Dor, who initiated the re-validation of Grosch's law and Mendelson, who discovered the statistical danger of using subgroups.

3. Computer capacity versus speed

The appropriate definition of computer capacity may also play a role in the search for Grosch's law type of effects in computer prices. At the time Grosch's law was formulated, computer capacity was simply expressed in terms of the speed of the processor, represented in terms of the number of instructions per second that the processor was able to execute. Typically, the studies of Knight [KNIGHT 1966] still refer to hundreds of instructions per second, which were replaced by the famous MIPS (millions of instructions per second) from the sixties on.

Actually, vendors and, subsequently the trade press, used the term "MIPS" but in fact used a form of relative performance to express the capacity of the computer system. This relative performance can be obtained from measurements on a benchmark by comparing the relative internal throughput ratio's. In fact , the basic laws of operational computer performance analysis [DENNING & BUZEN, 1978] confirm this very easily, as follows :

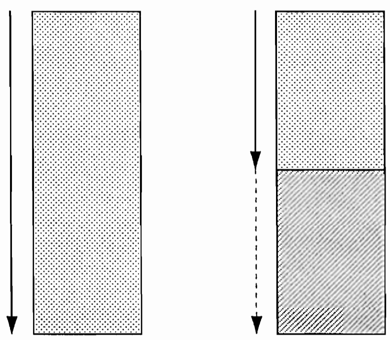
> Let A and B be two different computer systems. Running the same benchmark on both machines results in throughput values X_A and X_B and system utilizations U_A and U_B respectively. According to the utilization law
>
> $$U = SX \text{ , or } S = U/X$$
>
> so that $(X_B/U_B)/(X_A/U_A) = S_A/S_B$. Observe that U/X gives the systems utilization per transaction, which results in :
>
> S_A/S_B = speed radio between systems A and B =
> (% utilization per transaction on A)/(% utilization per transaction on B)

The last expression is sometimes referred to as the "internal throughput ratio", since the throughput is related to the busy time only, and not to the wallclock time, as in the external throughput X. Observe that, since the definition makes use of the workload-dependent performance parameter X, the above definition leads to a different speed by workload-type. At most, a "MIPS" rating is then based on average speed ratio's.

Speed is not equal to capacity, as was emphasised by Kleinrock [KLEINROCK 1986] and Denning [DENNING 1985]. They defined capacity (or "power") of a system as the ratio of the external throughput and the response time:

$$\text{Capacity} = X/R$$

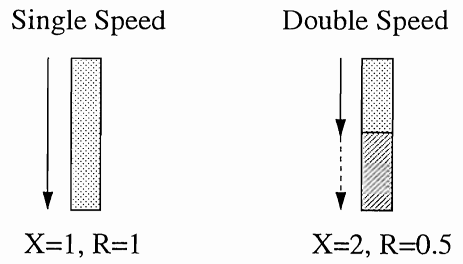
This definition acknowledges the fact that, within a single computer system, capacity can be used to produce throughput or response time, or a trade-off between both. This trade-off is of course economically determined, as the capacity allocation problem becomes a micro-economic problem of the optimization of a production function with one cost parameter (the capacity) versus two product parameters (throughput versus response time). In fact, without recognition of these economic aspects, the definition of capacity could easily lead to another Grosch's law illusion. A system with double speed can produce twice as much transactions with half the response time if that load is indeed available.



This could give a Grosch type of effect in the case where the external throughput X and the response time R have equal economic importance. In most applications however, only one of these parameters has a dominant value. Typically, administrative systems and operational systems have throughput as a determinator for economic value. The value of the system is, for example, directly related to the number of transactions per month that are processed. Decision support type of systems typically have response time as a dominant value indicator, since the speed of the decision is accelerated by the system (amongst other factors, such as the correctness of the decision). Many scientific jobs also belong to this group of applications, since a researcher typically wants to draw conclusions from the calculations or models that have been calculated by the job. Consequently, a more correct formulation (in analogy with the standard Cobb-Douglas production function) of the capacity of a system is the following :

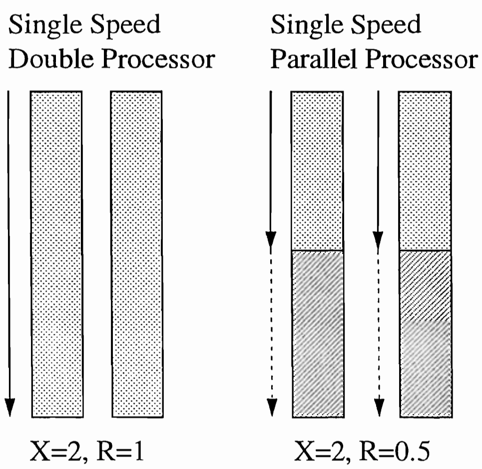
$$Capacity = X^a / R^b$$, where a/b gives the relative value of X over R ($a,b \geq 0$)

In fact, a/b gives the relative value of X over R, so that in the extreme cases (a or b = 0) the capacity is just proportional to X or $1/R$. This value determines to what extent the capacity will increase upon processor upgrade. The following examples illustrate that the use of this definition of capacity makes any conclusion about the existence of Grosch's Law impossible.



a=2, b=1	capacity~1	capacity~8
a=1, b=2	capacity~1	capacity~2.8284
a=0.5, b=0.005	capacity~1	capacity~1,4191
a=0.005, b=0.5	capacity~1	capacity~1,4191

It is interesting to revisit the arguments of Denning regarding the impact of multiprocessing and parallel computing on the relative capacity of computer systems. The following picture shows the performance data analogous to the above case, but this time for doubling the computer system by using two processors, in the first case just for multiprocessing, and in the second case for parallel processing :



In the multiprocessing case the relative capacity of the multiprocessor with respect to the mono-processor is 2^a (obviously neglecting effects of inter-processor overhead). The relative capacity of the double speed processor is $2^a 2^b$. This means that the capacity of a multi-processor can approach (economically) the capacity of a faster processor in case that the economic value of the throughput is largely dominant over that of the response time (b=0). It also means that for this type of applications a fair amount of "scaling" can be obtained with multiprocessing.

The case of a fully parallellisable job load on a multi-processor gives again a relative capacity of $2^a 2^b$, so that parallel processing can deliver the economic capacity of a faster processor. It should be stressed however that this only holds for a fully parallellisable workload, which is hard to achieve. In general, Amdahl's law on point accelerators applies [DEDENE & BURGER 1990] and shows that the general result of a partial speed-up is largely influenced by the portion that is accelerated. This makes parallel processing only an option in those cases where no faster speed can be obtained.

RAID disk processing is a typical application of parallel processing to the case of disk operations. Again, the effect of RAID disk processing on the average disk processing time will be proportional only to the amount of disk processing that it can speed up. For the same reason, many disk vendors combine RAID technology with disk caching. The use of multiprocessing for the disk I/Os gives at least the throughput increase effect, while the disk caching provides additional response time improvements. The combined result can give a substantial larger disk capacity (with the same "speed" of the individual disks).

The above arguments demonstrate that the notions of speed versus capacity should not be interchanged and can largely influence the discussion of possible Grosch effects.

4. Grosch's Law versus Moore's Law : exponential growth forever ?

The analysis of the previous sections brings in a relationship with another law in information technology : Moore's Law. This law, originally put forward by the Intel co-founder Moore in 1965, simply states that the *logic density of silicon integrated circuits in a chip doubles about every 18 months.* This law is strongly confirmed by the evolution of Intel chips in the last decade. Correlating density to cost, the arguments in the previous section would imply :

$$\textit{Moore's Law} \Rightarrow \textit{Grosch's Law}$$

Notice here that the actual cost degradation rate as stated in Moore's law is higher than the one in the simulation model of the previous section.

Obviously, the converse relationship is not necessarily true. First of all, the laws of physics reveal that Moore's Law cannot evolve continuously for eternity. In fact, it can be assumed that the creation of higher density gradually increases the cost of producing the actual chip. This of course slows down the degradation rates and makes it take more than 18 months to double the density at feasible prices. As such, there will be a point where the density increase drives this cost up to such a level that any cost advantage disappears. Some people argue that this effect may soon take place (period 2003-2005). From that point on, it makes no longer sense to make smaller chips. Of course, this puts an end to the validity of Moore's law at the same time. However, due to market expansion, prices can still go down for the same capacity and in this way still give rise to a Grosch like effect. So Grosch effects can still persist even when Moore's law stops.

Strange enough, as memory chips (also subject to Moore's Law) evolve in density, that density gets absorbed immediately. In fact, Parkinson's law of data (*buying more* *memory encourages the use of more memory-intensive techniques*) states that over the last ten years the memory usage of evolving systems tends to double about once every 18 months, which is precisely in line with Moore's law. However, memory chips are a nice example of the fact that the price doesn't follow the evolution of the densities : prices for memory chips have gone up with 10% on average on a yearly basis.

Last but not least, Moore's law type of effects are observed in networking [STEHLO 1995], in particular with the advent of ATM. As prices follow the increase in capacity, again a Grosch-like effect may seem to encourage the acquisition of the highest speed connections.

5. Conclusion

The conclusions from this paper are clear : Grosch's law was and is a statistical illusion. Its validation was (wrongly) based on a regression analysis through data series subject to decreasing computer prices, analogous to Moore's law. Furthermore, every technology that is subject to Moore's law type of effects can give rise to further illusions of the existence of Grosch's law, depending on the cost structure of the technology. Modern definitions of computer capacity, expressed by power and speed could also give rise to Grosch-like effects if the appropriate micro-economic features are not taken into consideration.

Acknowledgement

The authors would like to acknowledge Laurent Dupon, who already in 1989 performed initial validation experiments in the context of his graduation thesis. Discussions with MBA-students in the "Economics of data processing" class, where this material was further developed, were also greatly appreciated.

References

(Bernstein, 1987)
Bernstein Research Report, *The Myth of Price/Performance*, May 1987.

(Cale, 1979)
Cale E.G. et al., *Price/Performance of U.S. Computer Systems*, Communications of the ACM 22, 4 (1979).

(Dedene & Burger, 1990)
Dedene G. and E. Burger, *The Economics of Point Accelerators*, Proceedings of IEEE EDAC 90, IEEE Computer Press (1990).

(Denning, 1985)
Denning P.J., *Parallel Computation*, American Scientist , vol 73 (1985).

(Denning & Buzen, 1978)
Denning P.J. and J.P. Buzen, *The Operational Analysis of Queueing Network Models*, ACM Computing Surveys, Vol. 10, No. 3, September 1978.

(Ein-Dor, 1985a)
Ein-Dor P., *Grosch's Law Re-visited : CPU Power and the Cost of* Computation, Communications of the ACM 28, 2 (1985).

(Ein-Dor, 1985b)
Ein-Dor P., *Rejoinder to the Response on Re-revisitation (of Grosch's Law)*, Communications of the ACM 28, 7 (1985).

(Gilder, 1994)
Gilder G., *Auctioning the Airwaves*, Forbes ASAP, April 11, 1994.

(Grosch, 1953)
Grosch H.A., *High Speed Arithmetic : the digital computer as a research tool*, J. Opt. Soc. Am., Vol. 43, No.4, April 1953.

(Grosch, 1975)
Grosch H.A., *Grosch's Law revisited*, Computerworld 16/04/1975.

(Henkel, 1981)
Henkel T., *Hardware Roundup. A Look at 98 Systems From 17 Vendors,* Computerworld 13/06/1981.

(Henkel, 1988a)
Henkel T., *Hardware Roundup. Large and Medium Scaled Systems,* Computerworld 19/09/1988.

(Henkel, 1988b)
Henkel T., *Hardware Roundup. PC's, Workstations and Small Systems,* Computerworld 13/10/1988.

(Jones, 1985)
Jones J., *On Grosch's Law Revisited*, Communications of the ACM 28, 5 (1985).

(Kang, Miller & Pick, 1986)
Kang Y.M., Miller R.B. & Pick R.A., *Comments on "Grosch's Law Re-revisited : CPU Power and the Cost of Computation*, Communications of the ACM 29, 8 (1986).

(Kleinrock, 1986)
Kleinrock L., *Comments on "Grosch's Law Re-revisited : CPU Power and the Cost of Computation*, Communications of the ACM 29, 8 (1986).

(Knight, 1966)
Knight K.E., *Changes in computer performance*, Datamation (1966).

(Knight, 1968)
Knight K.E., *Evolving computer performance*, Datamation (1968).

(Mendelson, 1987)
Mendelson H., *Economies of scale in computing : Grosch's Law Revisited*, Communications of the ACM 30, 12 (1987).

(Serlin, 1986)
Serlin O., *Mips, Dhrystones and other Tales*, Datamation (1986).

(Stehlo, 1995)
Stehlo K., *ATM will overthrow Moore's Law*, Datamation, vol. 41 nr. 11, june 15, 1995, pp 7-8.

Appendix

The following is the SAS program for the simulation experiment.

```
options pagesize=55 pageno=1 nodate;

title1 'Simulation';

data grosch(keep=mips avcost logw logp);
   retain seed 98745632;                       /*seed random numbers*/
   years=9;                                    /*number of years    */
   mmips=15;                                   /*max cap year 1     */
   cpm=550;                                    /*cpm during year 1  */
   r=2;                                        /*shape parameter    */
   do year=1 to years;
      do i=1 to 20;                            /*take 20  samples   */
         call rangam(seed,r,t);
         mips=mmips*t*2/7;                     /*sampled # mips     */
         call ranuni(seed,s);
         call ranuni(seed,v);
         avcost=cpm-(s*60)+(v*60);             /*sampled cpm value  */
         logw=log(mips);
         logp=log(avcost);
         if mips <= mmips and avcost > 0 then
            do;
               output grosch;
            end;
      end;
      mmips=mmips*1.35;                         /*cap increase 35%   */
      cpm=cpm*0.75;                             /*cpm decrease 25%   */
   end;
run;

proc plot data=grosch;
   plot avcost*mips="*" /vaxis=0 to 650 by 100 haxis=0 to 90 by 10;
run;

title3 'regression';

proc reg data=grosch;
   model logp=logw;
run;

data grosch1;
   set grosch;
   intw = int(mips+0.5);
run;

proc sort data=grosch1;
   by intw;
run;

proc means data=grosch1 noprint;
   by intw;
   var avcost;
   output out=grosch2 n=aantal min=min max=max mean=gem;
run;

data grosch3;
   merge grosch1 grosch2;
   by intw;
run;
```

```
proc sort data=grosch3;
   by descending intw descending avcost;
run;

data grosch3;
   set grosch3;
   intwp=lag(intw);
run;

data grosch3;
   set grosch3;
   if intw < intwp;
   if intw > 0 ;
   logpm = log(max);
run;

title3 'regression : max';

proc reg data=grosch3;
   model logpm = logw;
run;
```