An Analysis of Intel CPU Pricing Data

Bob Drzyzgula

August 29, 2000

Abstract

The purpose of this analysis is to identify computer (Intel-processor PC) purchasing strategies that minimize the aggregate, long-term hardware costs required to obtain a given compute capacity. The primary inputs to this analysis are (a) official Intel pricing data for the years 1993-1999, and (b) officially-reported and estimated SPECint95 ratings for various Intel i86-architecture processors.

1 Background & Motivation

1.1 Overview

In this paper, I present an analysis of prices of Intel CPUs in relation to the integer processing performance of the same. In this analysis, I introduce a model for the performance over time of Intel processors at various pricing levels. This model, which appears to fit the data quite well, can be stated very concisely. For a pricing level b in dollars and a time t expressed in months since the beginning of the analysis period, I demonstrate that the performance $\rho_b(t)$, of the fastest Intel CPU selling for no more than price b at time t can fairly accurately be estimated by the function:

$$\hat{\rho}'_b(t) = (\rho_0 + sb)2^{\frac{t}{18}} \tag{1}$$

with ρ_0 and s being constants that are calculated through a least-squares regression.

Using this function, I calculate the effectiveness of various puchasing strategies in minimizing the cost to obtainin a required 12-year "compute capacity". Further along this line of calculations, I include a cursory examination of some of the non-CPU costs associated with providing this computing capacity, and identify two overall aquisition strategies which are more cost-effective than several others in providing one particular level of compute capacity.

This paper concludes with a tangentialy-related examination of the effectiveness of a single CPU purchase.

1.2 Context

In the Automation and Research Computing Section (ARC) of the Division of Research and Statistics at the Federal Reserve Board, we assemble our own PCs, in-house. There are myriad reasons for doing this, among them cost, control, flexibility, and the ability to provide superior support to our users. As one result of the flexibility that this strategy brings us, we are able to select component parts specifically for their reusability. Rather than dispose of an entire PC at the end of its useful life, we can usually reuse – at least once – parts that represent well over half of the original value of the PC.[1]

Far from this merely being the cheap, or "excessively frugal" thing to do, planning for reuse allows us to select parts that are of a higher quality and greater durability than we would be able to justify purchasing if we planned to dispose of them after two years. For example, we use top-quality power supplies, as well as tall, roomy, heavy-steel chassis with three ventilation fans, air intake filters and casters. Although these are expensive compared to more commonly-used chassis, the use of such high-end parts helps to keep our failure rate down and simplifies maintenance in the reduced number of instances where it is required. Even for parts that are not likely to be reusable, such as disk drives and motherboards, we usually find that purchasing unintegrated OEM parts in moderate quantities reduces the unit costs sufficiently that we can afford to use higher-quality, more reliable parts than are often found in "value" systems from volume manufacturers.

Moreover, we are able to provide more consistency for larger collections of devices. For example, with careful planning it can be possible to use the same model and firmware revision of disk drive in every PC deployed over an entire year. By purchasing large batches of parts such as motherboards, we can even stock very low-cost spare parts that are guaranteed to be absolutely identical – right down to the firmware revision – to parts which will ultimately need replacement. We have found that the maintenance-related labor savings resulting from this strategy more than offsets the labor costs associated with the initial assembly and more fragmented procurement.

1.3 Implications and Opportunities

In reflecting on the implications of and opportunities provided by this strategy, it becomes an interesting exercise to attempt to optimize the system upgrade and replacement strategy according to various criteria. Perhaps among the most interesting of these goals is the minimization of the cost to provide a given level of computing capacity over a period of time. By "compute capacity", I am referring to the time integral of one or more measure(s) of system performance. Thus, if one's system has performance measure ρ, and that system is held and used for time period $t_f - t_0$, the capacity of that system over that time period and according to this performance measure, would of course be $\rho(t_f - t_0)$. The performance measure does not have to be a measure of the rate at which the CPU can crunch numbers; it could be any measure of capability, such as disk storage, graphics polygons per second, etc.

If, however, during the same time period, one purchases replaces or upgrades that

[1]One important component that turns out to be reusable in this strategy is the operating system. While most commercially-sold Windows-based systems come with "OEM" licenses prohibiting the separation of the operating system license from the original hardware, retail licenses for Windows may be carried foward to a new hardware system if the old one is taken out of service. This means that, when we do a hardware upgrade, (a) we don't have to purchase a new Windows license, and (b) we are not forced to upgrade to a new version of Windows if the one we are currently using is no longer being offered.

system at regular intervals of duration Δt, then the total capacity of the the family of systems over that period would be

$$\mathcal{R} = \sum_{i=1}^{N} \rho_i \Delta t \tag{2}$$

where $\Delta t = \frac{t-t_0}{N}$ and ρ_i is the performance of the i^{th}unit held. Were it to be possible – or useful – to implement such upgrades continuously, this capacity figure would of course approach an integral$\int_{t_0}^{t} \rho(t)dt$. In reality, however, we will always be dealing with discrete sums, not integrals.

Of course, any upgrade to or replacement of the system will come at some cost. At the beginning of the i^{th} time period, some price π_i must be paid in order to gain the performance improvement $(\rho_i - \rho_{i-1})$. This price may bring either a whole new system, or perhaps only a small upgrade to an existing system, for example a new CPU and/or motherboard. Note that there is no magical assurance that the price paid will result in the performance increment being positive; this will only be true provided that, at each upgrade, a sufficient portion of the the system will be replaced so as to have such a favorable result. In addition, in order to make proper comparisons among dollar amounts spent over several years, it is necessary to apply a discount rate to these amounts to adjust them to a constant dollar value. Toward this end, let δ_i be the discount rate necessary to adjust dollars at the beginning of time period i to the value of dollars at a fixed point in time; in the calculations done later in this paper, I adjust all the dollars to values the end of the analysis period. Thus, the total amount spent over the time period $(t_f - t_0)$ will be:

$$\mathcal{P} = \sum_{i=1}^{N} \pi_i \delta_i \tag{3}$$

With these two values defined, it is straightforward to arrive at a measure of the effectiveness of the upgrade strategy in providing the compute capacity in question. The effectiveness $\mathcal{E}$ of this strategy is defined as "capacity per dollar" for systems purchased according to the strategy over the time period for the analysis:

$$\mathcal{E} = \mathcal{R}/\mathcal{P} \tag{4}$$

It is this measure $\mathcal{E}$ that we will attempt to maximize through a careful selection of purchasing strategy.

2 Source Data

The primary data source for this analysis is Intel's official, quarterly pricing announcements for i86-architecture processors from 1993 through 1999. In all, twenty-eight quarters of data are available for forty-eight different processors. This data is presented in Appendix A. Note that this paper considers only processors intended for use in mainstream, desktop PCs. For the period considered, this includes the 80486, Pentium (60-200MHz), Pentium MMX, Pentium Pro (256KB), Pentium II, Pentium III and the Celeron. This paper does *not* include data for the large-cache Pentium Pros or any of the Pentium II Xeon and Pentium III Xeon processors. Note that, while it would be interesting either to include or to do a separate analysis of AMD processors (AMD having consistently sold competitive or nearly-competitive, binary-compatible processors since the days of the 8086), I unfortunately know of no way to obtain comparable pricing or performance data for AMD's products.

The secondary source of data for this analysis is the SPECint95 benchmark. This benchmark, from The SPEC Consortium, `http://www.spec.org`, is a well-respected measure of the integer, or character, performance of a processor. Technically, it is a measure of system performance, but in practice the performance of a CPU dominates the measure and there is relatively little difference in performance among systems with the same CPU.

While official SPECint95 measurements are available for most of the CPUs considered in this analysis, there are several CPUs for which this measure is not available. In those cases, other benchmarks which measure integer performance were scaled to SPECint95 units based on the relative values for several CPUs for which both benchmarks are available. These other benchmarks were Intel's iComp, SPEC's SPECint92 and PC Magazine's Wintune Integer. In extending SPECint95 in this way, I was able to obtain a single performance metric for forty-eight different CPUs available over a seven-year period. To avoid confusion, when referring to my extended metric, I will use the the notation: "$Si95$". My $Si95$ numbers are also listed in Appendix A, annotated with the sources of any numbers that are not official SPEC benchmarks.

3 Theory

3.1 Notation

First, a bit of a glossary to kick-start my notation:

- $\mathbb{C}$ = the set of CPUs considered in this analysis.
- Q_o = the number of quarters of empirical pricing data included in this analysis. As of this writing, this number is 28.
- p_{cq} = the official Intel price for CPU $c \in \mathbb{C}$ and quarter $1 \leq q \leq Q_o$. In quarters where Intel did not have an official price for a CPU, this price is considered to be infinite.
- S_c = the $Si95$ of CPU c.
- b = a scalar price cut-off, or "bin", in dollars.
- $\mathbb{C}_{bq}$ = the set of $c \in \mathbb{C}$ where $p_{cq} \leq b$ in quarter $1 \leq q \leq Q_o$.
- $\rho_{bq} = \max(S_c)$ where $c \in \mathbb{C}_{bq}$. This can be described as the "$Si95$ of the fastest CPU costing no more than price b in quarter q".
- t = a continuous time in units of months, with t_0 the starting time for the planning period and t_f the ending time.

3.2 Moore's Law Curves

3.2.1 Background

As most everyone knows, Gordon Moore, in 1965, observed that the transistor densities in integrated circuits were growing and would continue to grow at a rate such that they would double every two years. Since that time, this prediction has been revised and extended, and eighteen months has become the generally-accepted doubling time for factors related to semiconductor performance.[2] Beyond just transistor density, this doubling time describes very many other measurable properties of integrated circuits, which generally change in direct relation to transistor density. For example, the more densely packed the transistors in an IC, the less the signal propagation delay between transistors, and the more quickly the circuits can switch on and off. The more quickly they can switch, the higher the clock rate at which the IC can operate, and the more work they can do in a unit of time. Thus, as a consequence, the growth rate of the measurable performance of CPUs tends to follow a Moore's law curve fairly closely.

Still, it is not immediately apparent that the performance at a given pricing level would follow this pattern. This only follows from the secondary observation that Intel has historically offered their latest, fastest CPU at a price in the range of $800-$1200. After introduction, Intel typically will allow the prices of their parts to soften in such a way that, by the time they are ready to introduce a new, yet-faster part, the prices of the older parts will have slipped so that they are less expensive than the newer parts in proportion to the relative speed differences between the old and new parts. This is less of a technological issue than a marketing issue. Intel's cost of manufacturing for these processors is extremely low when compared to the selling price and they could choose a different pricing strategy if it suited them.[3] They appear to have concluded that, under most conditions, profits will be maximized when they arrange for the price history for an individual part to follow a sort of inverse Moore's-law curve. By contrast, following the release of AMD's K6-2, Intel's processor prices were subjected to unprecedented downward pressure. As a result, Intel was, for almost a year, forced to abandon their traditional pricing strategy and allow their pricing structure to be compressed downward. Only after they introduced the "value-line" Celeron processors to bear the brunt of the price competition were they able to return to the old strategy in their flagship Pentium line.

While this connection is somewhat tenuous, given that we are attempting to predict something as vagarious as Intel's CPU pricing, this may in fact be close to as good as we can do.

---

[2] "An Update on Moore's Law" Gordon Moore, Intel Developer Forum Keynote September 30, 1997 `http://www.intel.com/pressroom/archive/speeches/gem93097.htm`

[3] It is also possibly a supply/demand issue, in that Intel may perhaps be selling these parts for the lowest price they can without creating a demand for a quantity of parts that is substantially greater than what they are able to manufacture. If this is true, then were Intel to lower the prices of their CPUs, price might well rise all by itself, but in such a way that the margin would belong to the resellers and speculators rather than Intel.

3.2.2 Estimation

What I will do here is to calculate a family of Moore's-law curves, each of which will estimate the performance of Intel CPUs vs time for a given fixed price ceiling. Specifically, for each price cut-off b, I will calculate a function:

$$\hat{\rho}_b(t) = \hat{\rho}_b(t_0)2^{\frac{(t-t_0)}{18}} \tag{5}$$

Note that, with the doubling rate fixed at eighteen months, each such curve is completely described by a single scalar parameter $\hat{\rho}_b(t_0)$, and thus the task is reduced to calculating this single coefficient for each price b of interest. To do this, I used a simple least squares estimation. Taking the log of both sides of equation 5, we have

$$\ln \hat{\rho}_b(t) \quad = \quad \ln \hat{\rho}_b(t_0) + \frac{\ln 2}{18}(t - t_0) \tag{6}$$

Setting $\hat{r}_{b0} = \ln \hat{\rho}_b(t_0)$, $r_{bq} = \ln \rho_{bq}$ and $\tau = (3(q-1) - t_0)$ (the $3(q-1)$ term serving to convert the quarter index to month units) then at each quarter q, the error in this estimate will be

$$r_{bq} - \hat{r}_{b0} - \frac{\ln 2}{18}\tau \tag{7}$$

and the sum of the squares of the errors will be

$$\sum_{q=1}^{Q_o}\{r_{bq} - \hat{r}_{b0} - \frac{\ln 2}{18}\tau\}^2 \tag{8}$$

Differentiating with respect to $\hat{r}_{b0}$ and setting the derivative equal to zero gives us

$$0 \quad = \quad \sum_{q=1}^{Q_o} 2\{r_{bq} - \hat{r}_{b0} - \frac{\ln 2}{18}\tau\} \tag{9}$$

$$\sum_{q=1}^{Q_o} \hat{r}_{b0} \quad = \quad \sum_{q=1}^{Q_o}\{r_{bq} - \frac{\ln 2}{18}\tau\} \tag{10}$$

$$\hat{r}_{b0} \quad = \quad \frac{1}{Q_o}\sum_{q=1}^{Q_o}\{r_{bq} - \frac{\ln 2}{18}\tau\} \tag{11}$$

$$\hat{\rho}_b(t_0) \quad = \quad \exp\left\{\frac{1}{Q_o}\sum_{q=1}^{Q_o}\{\ln \rho_{bq} - \frac{\ln 2}{18}(3q - 3 - t_0)\}\right\} \tag{12}$$

Using this formula, I calculated a $\hat{\rho}_b(t_0)$ for $t_0 =$ February 1993 and ten values of b: \$150, and \$200 through \$1000 in \$100 increments. The results are presented in table 1. The r^2 values listed in this table are for the the result of equation 5; they measure the correlation between $\hat{\rho}_b(3(q-1))$ and ρ_{bq}.

b	$\hat{\rho}_b(t_0)$	r^2
$150	0.6053	0.975
$200	0.6873	0.971
$300	0.8010	0.989
$400	0.8758	0.977
$500	1.0193	0.976
$600	1.1479	0.950
$700	1.2308	0.978
$800	1.3055	0.974
$900	1.4627	0.970
$1000	1.5336	0.969

Table 1: Moore's Law Coefficients

4 Practice

4.1 Strategies and Simulation

With these curves in hand, it is now much more straightforward to estimate the effectiveness factor $\mathcal{E}$ for a given strategy. Toward that end, define a strategy $\mathcal{A}$ to be a planned series of purchases $\{a_1, a_2, ...a_N\}$ at times $t(a_i)$, with the planning period being defined as times t_0 through t_f and subject to the constraints $t(a_1) = t_0$ and $t(a_N) < t_f$. At each purchase a_i, we will spend $\pi_B(a_i)$ on the base system components (chassis, power supply, and other parts with relatively long useful lives), $\pi_M(a_i)$ on moderate-lifetime parts such as the motherboard and memory, and $\pi_C(a_i)$ on the CPU itself. Thus

$$\pi(a_i) = \pi_B(a_i) + \pi_M(a_i) + \pi_C(a_i) \tag{13}$$

Note that, at each purchase, one or more of the price components may be zero. For example, we might purchase only a CPU upgrade, leaving the remainder alone, so that $\pi_B(a_i) = \pi_M(a_i) = 0$ and $\pi_C(a_i) > 0$. Or, more likely, we might purchase a new CPU, motherboard and memory, in which case, $\pi_B(a_i) = 0$ while $\pi_C(a_i)$ and $\pi_M(a_i)$ are both positive. In any event, each purchase will result in a system with performance $\rho(a_i)$. It is assumed that, at a minimum, each purchase includes a new CPU.

To simplify the calculations, two additional assumptions are made here. First, we assume that the "performance" of interest here is completely determined by the rated $Si95$ of the CPU in use following the purchase. Second, we assume that the persons doing the purchasing exercise a reasonable amount of common sense, and ensure that the system resulting from each purchase is able to take full advantage of the processor that is installed following the purchase. Thus, if newer, a faster motherboard and memory are necessary to take good advantage of a new processor, then those will be purchased. Clearly this will have to be done at some regular basis, since CPU interfaces change reasonably often and it thus will frequently be the case that any new CPU available will generally be unusable in most older systems.

As a result, the value $\rho(a_i)$ can be estimated as:

$$\hat{\rho}(a_i) = \hat{\rho}_{\pi_C(a_i)}(t(a_i)) \tag{14}$$

Defining a new function $\Delta t(a_i)$ as

$$\Delta t(a_i) = \begin{cases} t(a_{i+1}) - t(a_i) & , \quad 1 \leq i < N \\ t_f - t(a_N) & , \quad i = N \end{cases} \tag{15}$$

equation 2 may be estimated as

$$\hat{\mathcal{R}}(\mathcal{A}) = \sum_{i=1}^{N} \hat{\rho}(a_i)\Delta t(a_i) \tag{16}$$

And equation 3 may be estimated as

$$\hat{\mathcal{P}}(\mathcal{A}) = \sum_{i=1}^{N} \pi(a_i)\delta(t(a_i)) \tag{17}$$

where $\delta(t(a_i))$ is the discount rate at time $t(a_i)$. Finally, the effectiveness of a strategy $\mathcal{A}$ may be estimated as

$$\hat{\mathcal{E}}(\mathcal{A}) = \hat{\mathcal{R}}(\mathcal{A})/\hat{\mathcal{P}}(\mathcal{A}) \tag{18}$$

4.2 A Simple Observation

Thinking a bit about equation 18, it is of course obvious that the effectiveness will be improved by spending less money on faster systems. In the real world, it is also of course true that, all else being equal, the fastest systems tend to cost the most money, and the slowest systems tend to cost the least. Looking just at CPUs for a moment, we can see that, at any point in time, there is nearly a linear relationship between CPU cost and performance. Referring back to table 1, we can see pretty much exactly this behavior. Stunningly, the coefficients $\hat{\rho}_b(t_0)$, plotted against the values of b, form almost exactly a straight line; with $r^2 = 0.996$,

$$\hat{\rho}_b(t_0) \cong \hat{\rho}'_b = 0.463 + 0.00109b \tag{19}$$

For convenience, I will define $\rho_0 = 0.463$ and $s = 0.00109$, so that

$$\hat{\rho}'_b = \rho_0 + sb \tag{20}$$

This is so useful in simpifying the calculations that follow and the fit is so good, that I use this estimate $\hat{\rho}'_b$ in place of $\hat{\rho}_b(t_0)$ for the remainder of this paper.

4.3 Back to the simulation

Returning to the calculation of the effectiveness of various strategies, we can now simplify equation 16 by using equation 20 as an approximation for the coefficients. Specifically, we can now write

$$\hat{\mathcal{R}}'(\mathcal{A}) = \sum_{i=1}^{N} (\rho_0 + s\pi_C(a_i)) 2^{\frac{t(a_i)}{18}} \Delta t(a_i) \tag{21}$$

At this point, we need to consider some actual strategies, of which there are of course far to many to count. To keep things on a calculable and conceptually understandable level, we will make yet more simplifying restrictions on our choice of strategies. First, we will only consider strategies for which $\pi_C(a_i)$ is a fixed value $\pi_C(\mathcal{A})$ for all purchases within a given strategy. Second, we will only consider strategies for which the time between purchases $\Delta t(a_i)$ is again a fixed value $\Delta t(\mathcal{A})$ for the strategy. Moreover, we will limit $\Delta t(\mathcal{A})$ to be (a) an integral number of quarters, (b) evenly divisible into our planning period, which will be twelve years, or forty-eight quarters, and (c) no more than three years in length. Thus, the only values for $\Delta t(\mathcal{A})$ that will be considered are 3, 6, 12, 18, 24 and 36 months.

To begin this analysis, first observe that in calculating $\hat{\mathcal{E}}(\mathcal{A}) = \hat{\mathcal{R}}(\mathcal{A})/\hat{\mathcal{P}}(\mathcal{A})$, since the value $\hat{\mathcal{R}}'(\mathcal{A})$ does not reference purchases of anything but CPUs, the numerator cannot be increased or decreased by any changes in planned purchases for motherboards, disk drives, etc. Secondly, observe that, compared to purchasing only CPUs, purchasing these other components as well can only *increase* the denominator. Thus, it will prove useful to fist calculate the costs and effectiveness of several CPU purchasing strategies in isolation from the rest of the system parts. This having been done, we will consider several strategies for purchasing the remainder of the system parts, and investigate the impact each of these will have on the various CPU strategies.

With these simplifications, equation 21 now can be written

$$\hat{\mathcal{R}}'(\mathcal{A}) = \sum_{i=1}^{144/\Delta t(\mathcal{A})} (\rho_0 + s\pi_C(\mathcal{A})) 2^{\frac{(i-1)\Delta t(\mathcal{A})}{18}} \Delta t(\mathcal{A}) \tag{22}$$

$$= \{\rho_0 + s\pi_C(\mathcal{A})\} * \left\{ \Delta t(\mathcal{A}) \sum_{i=1}^{144/\Delta t(\mathcal{A})} 2^{\frac{(i-1)\Delta t(\mathcal{A})}{18}} \right\} \tag{23}$$

The two major portions of this last equation are easily calculable, and table 2 contains values of the coefficient $\{\rho_0 + s\pi_C(\mathcal{A})\}$ for the ten usual prices (note that these numbers should be highly similar to those in table 1), while table 3 contains values for the second factor. Table 4 shows the products of these two sets of values Referring especially to table 4, there are a number of observations that can be made. Foremost among them is that (duh) total capacity $\hat{\mathcal{R}}'$ may be increased either by purchasing more expensive CPUs, and/or by buying them more frequently. Secondly, it becomes clear that there are some capacities that cannot be achieved except by purchasing CPUs more frequently. For example, it would not be possible to satisfy a requirement for a total, 12-year capacity of over 7,000 $Si95$-months while purchasing CPUs only every 24

π	$\rho_0 + s\pi$
\$150	0.627
\$200	0.681
\$300	0.790
\$400	0.899
\$500	1.008
\$600	1.117
\$700	1.226
\$800	1.335
\$900	1.444
\$1000	1.553

Table 2: Values of $\rho_0 + s\pi$ for several interesting values of π.

Δt	$\Delta t \sum_{i=1}^{144/\Delta t} 2^{\frac{(i-1)\Delta t}{18}}$
3	6,250
6	5,890
12	5,210
18	4,590
24	4,030
36	3,060

Table 3: Values of $\Delta t \sum_{i=1}^{144/\Delta t} 2^{\frac{(i-1)\Delta t}{18}}$ for six values of Δt

$\frac{\Delta t \Rightarrow}{\pi \Downarrow}$	3	6	12	18	24	36
\$150	**3,910**	**3,690**	3,260	2,880	2,520	1,920
\$200	4,250	4,010	**3,550**	3,130	2,740	2,080
\$300	4,940	4,650	4,120	**3,630**	3,180	2,420
\$400	5,620	5,290	4,680	4,130	**3,620**	2,750
\$500	6,300	5,930	5,250	4,630	4,060	3,090
\$600	6,980	6,580	5,820	5,130	4,500	3,420
\$700	7,660	7,220	6,390	5,630	4,940	**3,750**
\$800	8,340	7,860	6,960	6,130	5,380	4,090
\$900	9,020	8,500	7,520	6,630	5,820	4,420
\$1000	9,700	9,140	8,090	7,130	6,250	4,750

Table 4: Values of $\hat{\mathcal{R}}' = (\rho_0 + s\pi)(\Delta t \sum_{i=1}^{144/\Delta t} 2^{\frac{(i-1)\Delta t}{18}})$ for six values of Δt and ten values of π. Cell values have units of $Si95$-months

months or more, even if $1000 CPUs were employed. But generally, it would seem that given a requirement for, say, a 12-year capacity of 3,500 $Si95$-months (and how one arrives at a figure like this is *far* beyond the scope of this paper), one can choose from any of the 48 (out of 60 total) strategies that meet this requirement. Most likely, however, the most cost-effective choice will be one of six strategies, the strategies in each column that offer the least amount of capacity greater than 3500 $Si95$-months. This is because, while there may be other factors would favor one purchase frequency over another, it is unlikely that purchasing a more-expensive CPU than necessary will result in a more cost-effective strategy. These six strategies are bolded in table 4. The next part, of course, is how to choose.

5 Dollars

5.1 CPU Costs, and the Effectiveness Thereof

Since the whole point of this paper is the minimization of the cost to provide a given level of compute capacity, the primary criterion here is clearly cost. This is not to say that, ultimately, one would not temper this with other factors, such as the labor required to implement the various strategies, but, all else being equal or even comparable, cost is the overriding factor. This paper will, however, only consider the acquisition costs associated with the purchase of the hardware itself. An analysis of labor, maintenance, and other support costs is, again, beyond the scope of this paper.

As mentioned in section 4.3, we will first consider the cost of the CPU itself, and will look at the other hardware costs as a second step. From equation 17, we know that the adjusted cost to procure all the CPUs in a strategy is equal to $\sum_{i=1}^{N} \pi(a_i)\delta(t(a_i))$. At this point we need to have a discount rate of some sort. The purpose of this discount rate is to adjust for the opportunity lost by spending the money, and is set at an amount that reflects the interest that could be earned on the money if it was not spent. In the case of the Federal Reserve Board, the most directly applicable rate would appear to be the interest rate paid by Treasury securities. While a fully-careful analysis would use the actual and carefully-projected T-bill rates, this isn't *that* careful an analysis, so I'm just going to pick a fixed rate for the entire 12-year period: 1.75% per quarter, which is equivalent to an annual rate of about 7.19%. At this rate, the discount rate to be used in the equations is

$$\delta(t(a_i)) = 1.0175^{\frac{t_f - t(a_i)}{3}} \tag{24}$$

Applying the simplifications from section 4.3, this can be rewritten as

$$\delta(t(a_i)) = 1.0175^{\frac{144-(i-1)\Delta t(\mathcal{A})}{3}} \tag{25}$$

Δt	$\sum_{i=1}^{144/\Delta t} 1.0175^{\frac{144-(i-1)\Delta t}{3}}$
3	75.6
6	38.1
12	19.4
18	13.1
24	10.0
36	6.91

Table 5: Summed discount factors

$\frac{\Delta t \Rightarrow}{\pi \Downarrow}$	3	6	12	18	24	36
\$150	**\$11,300**	**\$5,720**	\$2,910	\$1,970	\$1,500	\$1,040
\$200	\$15,100	\$7,620	**\$3,880**	\$2,630	\$2,010	\$1,380
\$300	\$22,700	\$11,400	\$5,820	**\$3,940**	\$3,010	\$2,070
\$400	\$30,200	\$15,200	\$7,750	\$5,260	**\$4,010**	\$2,770
\$500	\$37,800	\$19,100	\$9,690	\$6,570	\$5,010	\$3,460
\$600	\$45,300	\$22,900	\$11,600	\$7,890	\$6,020	\$4,150
\$700	\$52,900	\$26,700	\$13,600	\$9,200	\$7,020	**\$4,840**
\$800	\$60,500	\$30,500	\$15,500	\$10,500	\$8,020	\$5,530
\$900	\$68,000	\$34,300	\$17,400	\$11,800	\$9,030	\$6,220
\$1000	\$75,600	\$38,100	\$19,400	\$13,100	\$10,000	\$6,910

Table 6: Total, adjusted 12-year costs $\hat{\mathcal{P}}(\mathcal{A})$ for six values of Δt and ten values of π.

allowing equation 17 to be re-written as

$$\hat{\mathcal{P}}(\mathcal{A}) = \sum_{i=1}^{144/\Delta t(\mathcal{A})} \pi_C(\mathcal{A}) 1.0175^{\frac{144-(i-1)\Delta t(\mathcal{A})}{3}} \tag{26}$$

$$= \pi_C(\mathcal{A}) \sum_{i=1}^{144/\Delta t(\mathcal{A})} 1.0175^{\frac{144-(i-1)\Delta t(\mathcal{A})}{3}} \tag{27}$$

Now, the sum in equation 27 can be straightforwardly calculated for the six values of Δt under consideration; the result of this calculation is provided in table 5.

Having calculated these values, it is then straightforward to calculate the total 12-year costs associated with these strategies. These costs are listed in table 6; the costs associated with the six candidate strategies marked in table 4 are bolded here.

Finally, we can divide table 4 by table 6, cell by cell, to obtain the effectiveness $\hat{\mathcal{E}}(\mathcal{A})$ for each of these strategies considered. The result is presented in table 7. Again, the effectiveness values for the six candidate strategies from table 4 are bolded. From this result, three strategies clearly stand out: \$200 CPUs every twelve months, \$300 CPUs every eighteen months and \$400 CPUs every twenty-four months. It is difficult to select from among these three, although from a strict cost-effectiveness measure, the \$300 @ 18 months strategy would clearly win.

$\frac{\Delta t\Rightarrow}{\pi\Downarrow}$	3	6	12	18	24	36
$150	**0.345**	**0.645**	1.120	1.460	1.680	1.850
$200	0.282	0.526	**0.915**	1.190	1.370	1.510
$300	0.218	0.407	0.708	**0.919**	1.060	1.170
$400	0.186	0.347	0.604	0.785	**0.902**	0.995
$500	0.167	0.311	0.542	0.704	0.809	0.892
$600	0.154	0.288	0.500	0.650	0.748	0.824
$700	0.145	0.271	0.471	0.612	0.703	**0.775**
$800	0.138	0.258	0.448	0.583	0.670	0.738
$900	0.133	0.248	0.431	0.560	0.644	0.710
$1000	0.128	0.240	0.417	0.542	0.624	0.687

Table 7: Effectiveness $\hat{\mathcal{E}}(\mathcal{A})$ for several CPU-only strategies

If one somehow had a high degree of confidence in the requirement for 3500 $Si95$-months as a strict upper bound on the usable capacity, then one would probably want to do a slightly different calculation; in that case, one would probably use 3500 as a constant value of $\hat{\mathcal{R}}'$ for any strategy that resulted in at least this much capacity, and divide this number by all the corresponding $\hat{\mathcal{P}}(\mathcal{A})$ values. In this case, it is obvious that lowest cost would be the controlling factor, and the $200 @ 12-month strategy would be regarded as optimal. Still, before a final selection can be made, the other costs associated with provisioning these systems must be examined.

5.2 Non-CPU costs and Overall Effectiveness of a Strategy

In addition to the CPU, a typical desktop PC needs a number of additional parts. These are presented in Table 8. While this table is instructive, we need to take some care in how to interpret it. It is not simply the case that the range between the “typical low cost” and “typical high cost” for these parts can generally be linked either to the cost of the CPU that is used or the length of time that the system will be used. Although these are factors for some items (e.g. one would want to spend more on a system chassis or keyboard if it needed to last longer, and some faster CPUs may require more expensive motherboards or memory), the cost of other items are likely to be controlled by special, localized requirements. For example, for basic Ethernet connectivity, a $20 network card may be entirely sufficient and likely to last for three years or more. However, if FDDI, ATM or fiber optic, Fast Ethernet connectivity are required in a particular installation, the network card costs may approach $250 even if the remainder of the system used bottom-end stuff. Thus, what I will do here is simply arive at a reasonably middle-of-the-road base price for parts that needed to last for 12 months, and then add a fixed amount for each additional six months that that parts needed to last. This rather extreme imprecision will allow us to limit the extent of this calculation for a single example, but still allow the process to be worked throuh to completion a single time. These figures are listed in Table 9. Note that, in this table, I have not included prices for either a 3-month or 6-month useful life. This is becuase I find it quite unlikely that

Item	Typ. Low Cost	Typ. High Cost	Category
System chassis & PS	$50	$300	B
Motherboard	$70	$300	M
Memory	$100	$400	M
Floppy or LS-120	$20	$60	B
CD/DVD-ROM	$50	$150	M
Hard Disk Drive	$90	$200	M
Video/Graphics Card	$50	$300	B
Network Controller	$20	$250	B
Sound Card	$20	$80	B
Keyboard	$10	$50	B
Mouse/Trackball	$10	$70	B
Speakers	$10	$100	B
“Base” Total	$190	$1210	B
“Moderate life” Total	$310	$1050	M
Total	$500	$2260	

Table 8: Parts, other than the CPU or monitor, used in a typical desktop PC. The “Category” column denotes whether this part is replaced along with the “Base parts” (B) or the “Moderate useful-life parts” (M).

$\frac{frequency\Rightarrow}{category\Downarrow}$	12	18	24	36
B	$700	$800	$900	$1100
M	$400	$600	$800	$1200

Table 9: PC part prices used for final analysis

B/M/C sched	B cost	M cost	C cost	$\hat{\mathcal{P}}$	$\hat{\mathcal{R}}'$	$\hat{\mathcal{E}}'$
12/12/12	$13,600	$7,750	$3,880	$25,230	3,550	0.141
18/18/18	$10,500	$7,890	$3,940	$22,330	3,630	0.163
24/12/12	$9,030	$7,750	$3,880	$20,660	3,550	0.172
24/24/12	$9,030	$8,020	$3,880	$20,930	3,550	0.170
24/24/24	$9,030	$8,020	$4,010	$21,096	3,620	0.172
36/12/12	**$7,610**	**$7,750**	**$3,880**	**$19,240**	**3,550**	**0.185**
36/18/18	**$7,610**	**$7,890**	**$3,940**	**$19,440**	**3,630**	**0.187**
36/36/36	$7,610	$8,300	$4,840	$20,750	3,750	0.181

Table 10: Overall costs, capacities and effectiveness factors for various 12-year strategies providing at least 3500 $Si95$-months of capacity.

it would be prove to be useful to replace a system chassis or disk drive, for example, every three or six months. About the only part that an argument could likely be made for replacing on this frequency is the CPU, and, given the resulting costs shown in Table 6, it seems clear that even then there would have to be a rather extreme requirement for the highest levels of performance available to justify such a strategy. In fact, in the remainder of the analysis, I will not consder replacing even CPUs any more frequently than once per year.

With these restrictions, there remain only a few valid scheduling strategies. These are are presented in Table 10. Costs are aggregate over the twelve-year planning period and are adjusted to end-of-planning-period dollars. The effectiveness value presented in this table is

$$\hat{\mathcal{E}}' = \hat{\mathcal{R}}'/\hat{\mathcal{P}} \tag{28}$$

That is, the effectiveness values are computed using the straight-line estimate $\hat{\rho}_b(t_0) \cong \rho'_b = \rho_0 + sb$.

In the end, then, at least in this one example, there is a single strategy that dominates all the others considered in effectiveness, while having the second-lowest 12-year adjusted cost: Purchasing a base system unit every 36 months for $1100, and purchasing a motherboard, memory, CPU, etc., every 18 months for $900. The closest alternative is the 36/12/12 strategy, which would have the CPU, motherboard, etc. replaced every tweleve months rather than every 18 months. This strategy is second-highest in effectiveness (0.185 instead of 0.187) but has the lowest overall cost. ($19,240 rather than $19,440). Ultimately, the choice between these two would probably need to be decided according to other criteria. The fact that the 36/12/12 strategy would require more frequent system assembly and disassembly is likely to be an important consideration.

I wish to emphasize here that the analysis presented in this section regarding non-CPU costs is just a single example of how one could procede in using the more complete analysis of CPU costs. Potentially, one could perhaps do as complete an analysis of non-CPU parts as has been done for CPUs in this paper. However, with the exception of system memory, it is likely that information in similar detail for these other parts is simply not available. Also, keep in mind that, in another situation, CPU performance may not be the factor of primary importance. In a server farm, for example, disk storage capacity might be much more important.

6 A Curious Calculation: The Effectiveness of a Single CPU Purchase.

Once in possesion of as simple a formula as equation 19, it is becomes somewhat tantalizing to tinker with it a bit and see where it might take us. In this section, I consider the effectiveness of a single purchase of a CPU – although most trade press articles will quote the *price-to-performance* of a CPU, I find it more useful to think in terms of *performance-to-price,* or *performance per dollar*, which is what I am calling *effectiveness* of the money spent on a CPU purchase.

For a CPU purchased at price b at time t (here, for simplicity, stated as an offset from t_0, and it is assumed that one would always purchase the fastest CPU selling for price b; clearly any other choice would result in a lower effectiveness value) we define the effectiveness $e_b(t)$ of that purchase as

$$e_b(t) = \frac{1}{b}\hat{\rho}'_b 2^{\frac{t}{18}} \tag{29}$$

$$= \frac{1}{b}(\rho_0 + sb)\, 2^{\frac{t}{18}} \tag{30}$$

$$e_b(t) = \left(\frac{\rho_0}{b} + s\right) 2^{\frac{t}{18}} \tag{31}$$

Clearly, at any time t, the effectiveness will vary with the inverse of the price paid – the higher the price paid, the lower the effectiveness. Note that, as b, the price paid, gets very large, the coefficient $(\frac{\rho_0}{b} + s)$ becomes dominated by s. That is, $s2^{\frac{t}{18}}$ becomes an asymptotic lower bound on the the effectiveness of any purchase at time t. Thus, there really is a strong law of diminishing returns at play here – beyond a certain price, the *relative* performance improvement one can purchase for an additional dollar is small compared to the improvement that is seen at the lowest prices.

One somewhat interesting observation that may or may not be related is that one rarely sees Intel (desktop) CPUs selling (as opposed to being offered) at prices above about \$850. Out of 309 data points in my price history, only 18 – about 6% – are greater than \$850 and in many of those cases it was questionable whether those parts were even actually for sale in any volume greater than a half-dozen or so; most processors that ever sold at that price (Pentium 60, 66, 100, 120, and 133; Pentium Pro 150, 180 and 200; Pentium II 300) only did so for one or two quarters, immediately following (a premature?) introduction. The only two processors that held such high prices for any significant time were the Pentium 66 and the Pentium 100, each of which remained above \$850 for four quarters following introduction, in May 1993 and February 1994, respectively. Interestingly, $b = \$850$ is exactly the point at which ρ_0/b is one half of s ($\rho_0/850 = 0.463/850 = 0.000545 = 0.5 * 0.00109 = s/2$). The point at which $\rho_0/b = s/4$ is around \$1,700 – out in the stratosphere of large-cache Xeons.

From this point, it is interesting to consider the loss in CPU purchase effectiveness which would result from changing a planned purchase to use a more highly-priced CPU than originally planned. If a CPU costing b_1 was planned, and this plan is changed to use a CPU selling at price $b_2 > b_1$, the loss in effectiveness will be

$$e_{b_1}(t) - e_{b_2}(t) = \left(\frac{\rho_0}{b_1} + s\right) 2^{\frac{t}{18}} - \left(\frac{\rho_0}{b_2} + s\right) 2^{\frac{t}{18}} \tag{32}$$

$$= \rho_0 2^{\frac{t}{18}} \left(\frac{1}{b_1} - \frac{1}{b_2}\right) \tag{33}$$

To put this in some perspective, it is probably more interesting to calculate the proportion of the effectiveness that is lost by making this change; this would be

$$\frac{e_{b_1}(t) - e_{b_2}(t)}{e_{b_1}(t)} = \frac{\rho_0 2^{\frac{t}{18}} (\frac{1}{b_1} - \frac{1}{b_2})}{(\frac{\rho_0}{b_1} + s) 2^{\frac{t}{18}}} \tag{34}$$

$$= \frac{(\frac{1}{b_1} - \frac{1}{b_2})}{(\frac{1}{b_1} + \frac{s}{\rho_0})} \tag{35}$$

Note that this result is not time-dependent, i.e., that the proportion of purchase effectiveness lost when moving from a less-expensive CPU to a more-expensive CPU is always the same, given identical choices for the two price points (or approximately the same, differing only as a result of momentary deviations from Intel's normal price strategy). Note also that the term s/ρ_0 is approximately 0.00235, which is about equal to $1/b_1$ for $b_1 = \$425$. Thus this term is significant in this calculation.

As an exercise, it is interesting to ask, given a base price b_1, at what value of b_2 is one-half of the effectiveness lost. In such a case,

$$\frac{1}{2} = \frac{(\frac{1}{b_1} - \frac{1}{b_2})}{(\frac{1}{b_1} + \frac{s}{\rho_0})} \tag{36}$$

$$2(\frac{1}{b_1} - \frac{1}{b_2}) = (\frac{1}{b_1} + \frac{s}{\rho_0}) \tag{37}$$

$$\frac{2}{b_2} = \frac{2}{b_1} - \frac{1}{b_1} - \frac{s}{\rho_0} \tag{38}$$

$$= (\frac{1}{b_1} - \frac{s}{\rho_0}) \tag{39}$$

$$\frac{1}{b_2} = (\frac{\rho_0 - b_1 s}{2 b_1 \rho_0}) \tag{40}$$

$$b_2 = \frac{2 b_1 \rho_0}{(\rho_0 - b_1 s)} \tag{41}$$

Table 11 contains the result of this for a few values of b_1. Note that beyond a certain value of b_2, it is not possible to lose half the purchase effectiveness, because of the observation made in section 6: there is a lower bound on the effectiveness of any purchase. The first value of b_1 for which it is not possible to halve the effectiveness can be deduced from equation 31. This occurs when $\rho_0/b_1 = s$, or $b_1 = \rho_0/s$, which, as mentioned previously, happens when $b_1 = \$425$.

b_1	b_2
\$150	\$464
\$200	\$756
\$300	\$2,043
\$400	\$13,719

Table 11: Prices (b_2) at which a CPU purchase loses half of it's effectiveness compared to a purchase at price b_1.