



Not Your Father's Lease: Lease versus Buy Decisions in a Technology-Intensive World

Introduction: Dependence on Technological Capacity

Today's modern organization exists in an operational environment that has become increasingly dependent on technology—not only to *thrive, excel, and exceed*, but even to *survive and adapt*. But, just as the dependency on technology has increased, so also has the *rate of change* of that technology. Gone are the days when an asset we placed into service in Year One still allowed us to be 'as competitive' as firms implementing current technology even two-three years later. The technology industry's ability to relentlessly produce more-capacity-for-less-cash (e.g., Moore's Law) creates competitive threats every new product cycle (e.g., 18 months, in most IT areas). Architectural choices have become increasingly difficult, as very few products remain capacity-competitive (with even their successors!) after as short as 12-24 months. Lease-versus-Buy decisions must incorporate **access-to-capacity** considerations and calculations into their analysis. Faster innovation generally entails faster obsolescence as well, and distributed assets (PC, mobile devices) are beginning to show extra support costs earlier than they did a decade ago. These TCO-type costs have to be included in refresh/financing planning. The old rules now have to share the harness with the new rules.

Centralized/Server Assets: The Capacity-based Refresh Principle

In most organizations, the finance function has two primary tasks: (1) obtain capital at the most favorable rates and terms, and (2) allocate this capital/capacity to initiatives which will convert this productive capacity into the greatest relative amount of value. This latter task is fundamental to the very purpose of finance, and **it is this principle that is most changed in its implementation today**.

A couple of decades ago, technology products were refreshed by the manufacturer in the 36-50 month range, with price increases from one generation to the next being the norm. Since most adopting organizations had technology growth demands much "lower and slower" than today's frantic pace, this scenario was reasonable. Products had very long usage cycles—4 to 8 years—because (1) the competitive marketplace was not producing attractive/compelling alternatives very quickly; and (2) the systems were built with over-capacity to begin with. This was not good for expense reasons, of course, since customers had to buy extra unnecessary capacity to get their required level (e.g., they had to buy a 'full box' to get access to the *25% of that box* they actually needed at the time). Fortunately, as product families began to produce entry-level and mid-level products, the capacity-needed versus capacity-purchased gap narrowed somewhat.

In that earlier period, the bare-numbers aspect of Lease-versus-Buy seemed simple: we totaled lease payments and compared this with the purchase price, perhaps adjusting for the time value of money (e.g., NPV). In most cases (barring other considerations such as tax, financial treatment, covenants, etc), the outcome was almost inevitable: buying was cheaper than 'renting' because of the long life of the asset. We added other factors to this comparison model—attempting to reflect cost of money, life-cycle costs, and admin costs—but at the end of the day, it was often still very favorable to 'buy' outcomes.

The main reason this process was more-or-less valid, of course, was that the dollars we were spending on the 'buy' were gaining us the same amount of capacity as the (larger amount of) dollars we were spending on lease payments. If "one unit of productive capacity" could either be bought for \$1,000 or

leased for \$1400 over a five-year period—and **neither approach varied the amount of capacity involved**—then, “all other things being equal,” purchasing was the fiscally clear choice.

In the early 1990’s however, things changed. Renewed interest in technology investment opportunities produced new industry entrants and new competitive market forces. IBM decided to get serious about challenging Sun in the UNIX server space, Intel decided to ‘appropriate’ customer dollars from entry-level UNIX servers to Intel-based servers, and high-end external disk storage vendors were *forced* to accelerate their product innovations—due to the rapid cost decay curve of hard drives (i.e., competitors could produce a 20-25% price advantage *within a year*, just by ‘waiting’ for prices of component disk drives to fall).

These forces accelerated (1) as competition became more intense, (2) as our need for this technology became more critical, and (3) as the costs of innovation declined (generally via wider operational leverage). The result of this—over about a 15 year period of churning—is this:

- ❑ CPU chip advances are measured in 6-9 month periods. Clock speeds, cache size, and bus speeds all improve incrementally *two to three times* every 12-18 months. Since these CPU’s are used in servers/hosts, they represent a way to increment capacity on a smoother curve;
- ❑ Major server/host product refreshes occur in 18-month cycles, with performance increases in the 25-40% range and price increases only in the 0-15% range (if any at all).
- ❑ This nets out to an average 25% price-performance increase *every 18 months*, distributed fairly smoothly across the 18 month period, and a 66% price-performance increase *every 36 months*.

In the centralized server market, this could allow an organization to match their **capacity need projections more closely with their capacity acquisition expenses**. For example, if we project that we will need to grow our technology capacity by 20-25% per year, over a smooth growth curve (instead of spikes in growth), then a technology refresh strategy which replaced servers with the newer, higher capacity models *more frequently* would best match expense to need. In this case, a dollar of expense—when refreshed every 9-18 months—would yield *incrementally* more capacity, in synch with our needs for that capacity. This is both a cost-avoidance strategy and a cash conservation strategy: the same dollar buys more capacity, but not more capacity than we need to have at any given moment (e.g. closer to a JIT model).

So, for firms enjoying (or ‘being challenged by’) growth rates in the 15-40% per annum range (in terms of capacity requirements, *not* revenue—revenue generally *lags* procurement of productive capacity, obviously), the most efficient way to address this is with faster-refresh approaches (e.g., moving from a 4-5 year refresh cycle to a 18-36 month cycle).

This faster-refresh approach applies mainly to entry-level and mid-tier servers/hosts, and less so to end-user technologies and highest-end servers/hosts. End-user technologies (desktop, mobile) now grow in ‘features’ instead of sheer performance (they have other reasons for shorter refresh cycles—see below); and high-end servers (e.g., mainframes and very large enterprise class Unix servers) have generally had a way to grow capacity *inside* the box, by **addition** of components. But entry/mid-tier servers are often limited in their ability to add new processors/components, requiring instead a ‘fork lift’ upgrade or **exchange** of components.

Notice too that this capacity-driven faster refresh is about **equipment replacement or exchange—not about additional units**. If we could simply buy a second box and harness it together with the first box into some kind of “cluster” (being able to run the two as an integrated unit, known as a ‘single system image’), then we could grow capacity incrementally *without* retiring the first box. In such a scenario, we would conceivably still be using equipment for those longer terms (i.e., 4-5 years), since there would be no incentive or requirement to decommission them. And, in such a case, the older dynamics of Lease-versus-Buy could still apply.

However, in today's technology environment, many/most of the important software applications we depend on **cannot run on such clustered systems**. The software manufacturers are scrambling to get there (it greatly expands their market if we customers could run their expensive software on clusters or grids of low-cost PC's, for example), but the number of mature offerings of this in mid-2006 is relatively minute. We are still locked into getting a new, single, higher-powered system (and decommissioning the older, lower-powered system from that workload) in order to grow capacity in this important technology area.

What this means is that **finance should be pushing for short (or shorter) refresh cycles in entry-level and mid-level servers**. Such a policy could dramatically increase capacity without comparable hardware expense, and better match expenses to equipment utilization.

Distributed/PC Assets: The Obsolescence/TCO-based Refresh Principle

Desktop and mobile (e.g. laptops, notebooks, PDA's, Smartphones) systems, however, manifest a different set of life-cycle characteristics.

They also get frequent performance upgrades from the vendors (often more frequently than servers, actually), but we are less convinced as managers that such incremental capacity is actually contributing *proportionately* to business performance. [A frequent complaint is that the additional power is 'squandered' on Web surfing and personal use of the PC while at work.] There are obvious cases where a refresh is required to bring our PC's "into the new world"—sometimes required for corporate and/or centralized applications—but otherwise assessing the business contribution of "an extra GigaHertz of speed" has generally eluded management science so far.

But, we have hit the ceiling on PC performance increases (i.e., clock speeds cannot appreciably increase due to heat considerations now), and the focus from the PC vendors is now on 'features'. These features are focused *somewhat* on end-user benefits (e.g., reliability, security, media), but *to a greater extent on IT managers* (e.g., better ability to manage, protect, trouble-shoot, and support the system). These technology advances are impressive and noteworthy, but the business benefit is (again) difficult to measure with confidence.

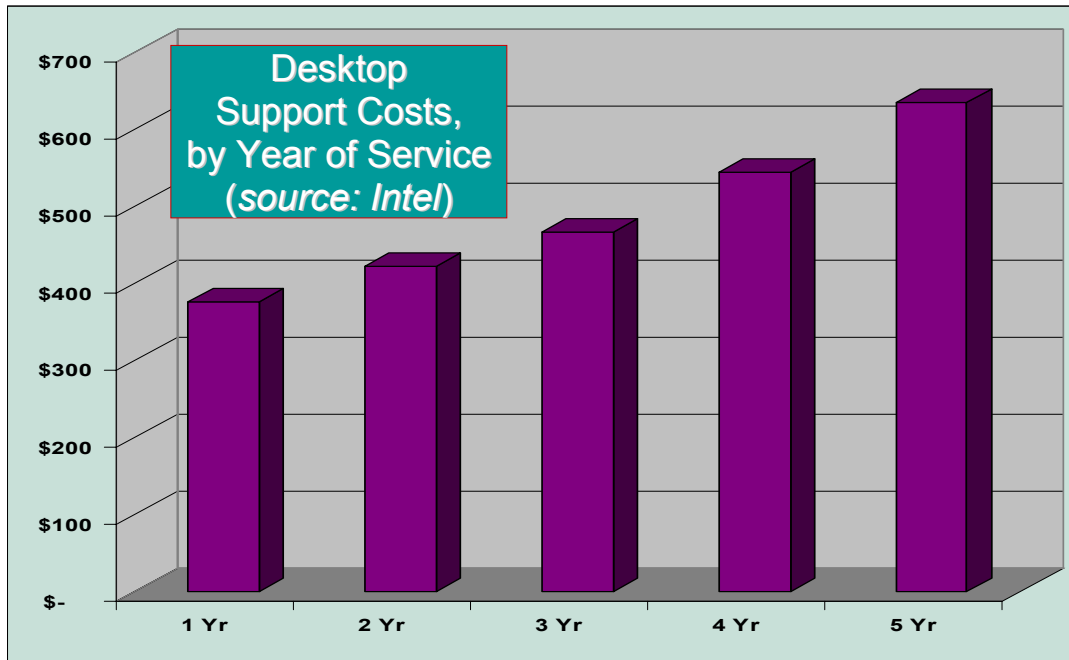
In the PC world, though, there are major discontinuities which must be planned for. New versions of Windows, new version of applications, and/or new versions of desktop management software sometimes require newer features in the PC. As we approach the introduction of Vista (the next version of Windows), for example, we already know that certain video/graphics chips in our existing systems will only support "sub-functionality" in that new release. Our processor speed might be fine, but without the right graphics card or wireless chip or hardware BIOS level we cannot have all the goodies. Fear of being 'left behind' by software advances has been a major argument for tech-refreshes in the PC/mobile device market for years—and a warranted one, in many cases.

Such reasons/arguments for refreshes have been notoriously difficult to quantify for ROI analysis, unfortunately, and many have just assumed desktop refreshes to be a 'cost of doing business' (like regulatory compliance). Others have agreed with this, but have also factored in the rising costs of secure-and-compliant disposal, a growing legal and PR concern.

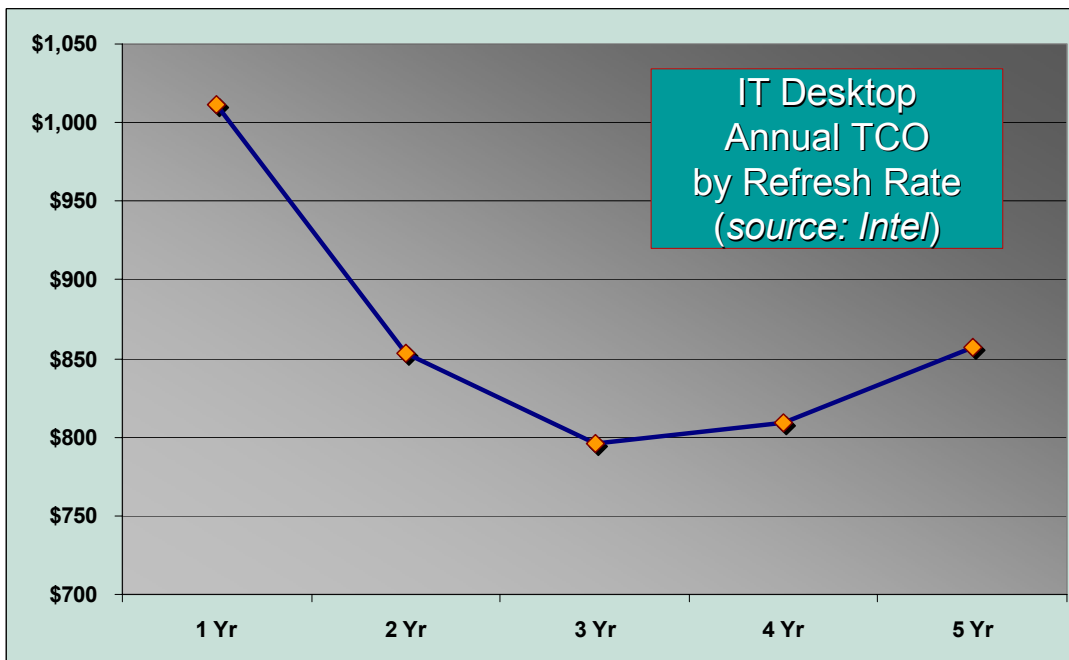
But new TCO considerations have begun to surface, tracked by the industry analysts and technology manufacturers. Intel's internal TCO study has recently caught public attention, and although we might *suspect* them of spinning the numbers to support faster uptake of their products (!), the numbers in the study seem realistic and sober. We do know that labor, repair, patching, and support costs increase *with each year of usage*.

And, in terms of refresh cycles, 3-year-old PCs are now **at least 2 product refreshes old**, with possible complications/costs in maintenance contracts, features availability from software upgrades, and compatibility with new corporate and commercial initiatives.

Intel's study tracked their internal costs for PC's over a 5 year period. Their findings support the general intuition that TCO costs begin to increase dramatically after 36 months. Here are two charts, one on support costs by year, and one on annualized TCO costs:



These support costs consisted of: Patching, Level 1 Helpdesk calls, Level 2 Helpdesk Calls, Out-of-Warranty Repairs.



Costs not included in these charts are any upgrades/enhancements required to run new software (e.g., Vista, .NET, corporate apps, memory-intensive apps).

How much incremental cost is associated with years 4 and 5? Looking at the support cost chart, we can see that the support costs for years 4 and 5 are an extra \$247 over what the support costs would have been had the support costs stayed constant at the 3-year mark [\$78 over in year 4 and \$169 over in year 5]. This represents an obsolescence cost (mostly labor, but out-of-warranty repairs are hard costs). Looking at the TCO chart, we can see that the increases in annual costs in years 4 and 5 amount to an extra \$52 and \$305, respectively, when totaled across the 4 and 5 year periods.

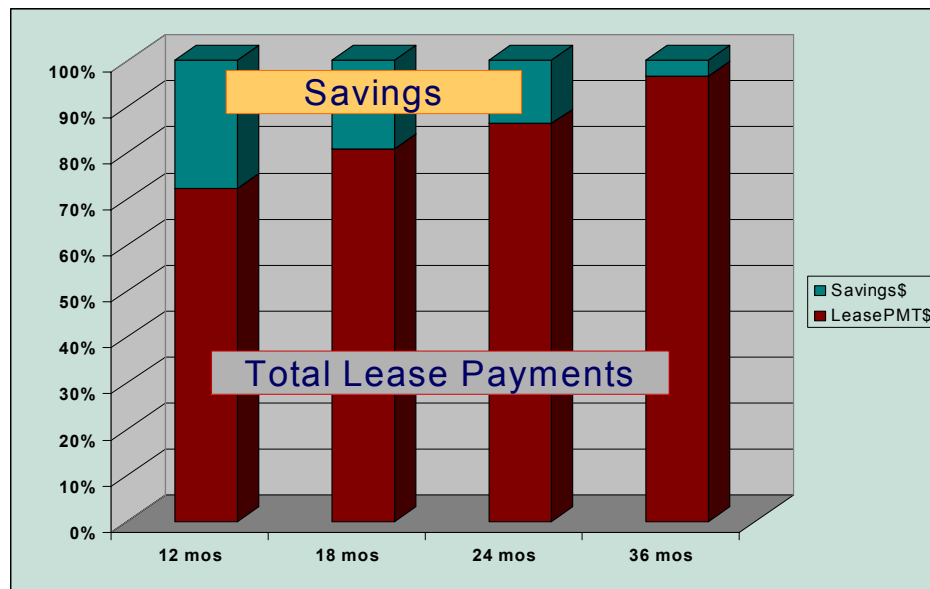
All of these figures are relative to a base cost of \$800 for the PC, and the TCO curve reflects that cost spread across the number of years (e.g. the 3-year annual cost point has only \$280 for acquisition cost). Some of the increases in support costs would be offset by the decrease in the amortized acquisition cost, but this is 'funny money': 1) the support costs represent real labor/payroll expense, hours that cannot be used for other gainful tasks, and 2) out-of-warranty repair costs are invoice-dollars and cannot be amortized (since there is no time 'left' for this, being at the end of the life cycle). Thus the Year 4 support cost 'overage' is approximately 10% of the purchase price, and the Year 5 'overage' is 20%. Keeping the PC for two years past the 36 month point thus generates an additional 30% cost, over the original purchase price.

So, **for distributed end-user assets** (and in some situations, for distributed entry-level x86 servers), **support/obsolescence cost realities would suggest that shorter refresh cycles are to be preferred**—36 months at the longest.

The Impact of Shorter Refresh Cycles on Lease-versus-Buy Decisions

When we move to refresh cycles in the 18-36 month range, **the traditional longer term Lease-versus-Buy economics are reversed: leasing always 'wins'**, and it wins 'more dramatically' the shorter the refresh cycle. There are **two methods to calculate/estimate this**: a cash-flow comparison and an administrative TCO comparison.

The cash-flow comparison simply compares the total purchase price against the sum of the lease payments (cost of funds can be added to the calculation as well). On a \$1,000 asset, for example, here is how the cash-flow comparison looks, by length of term:



The leasing rates used in this chart are based on 'lessor-friendly' terms and conditions (including credit worthiness, current technology, 'normal use'), but even if we reduce the savings by a third, we still can generate considerable savings per \$1,000 of assets deployed. [Notice also that the cash-flow savings at the 18-month refresh point would coincide with a price-performance improvement in the 20-30% range.]

An administrative TCO comparison would situate these cash-flow savings *in the context of life-cycle administrative costs*. This wider context of costs would include transition costs, disposal/remarketing, maintenance of an asset database, etc. Here is a side-by-side comparison of Lease-versus-Buy for a 36 month scenario:

	Full Purchase	Lease*	(Notes)
Invoice(s) Costs:	100.0%	88.0%	(1)
Asset Database Creation/Maint.:	4.0%		(2)
Lease Management:		1.0%	(3)
PC-to-PC Migration/Transfer:	5.0%	5.0%	(4)
De-installation:	2.5%	2.5%	(5)
Shrink/Warehousing:	5.5%		(6)
Freight:	2.5%	2.5%	(7)
Vendor Disposal Fees (Avg):	15.0%		(8)
Total Cost of Ownership:	134.5%	99.0%	

* 36 Month, Operating Lease

Notice that in the regular 'pay for disposal' scenario, **the cost spread between leasing and buying is almost 35%** (134.5% minus 99%).

This model excludes many of the *non-administrative* TCO costs, such as broader tech support, security, and network connection, but these operational costs are *identical* under both of our financial alternatives here. Each company will have different percentages to use in this simple model, but it is important to recognize **all** the invoice and payroll costs associated with the model. To that end, the notes in the table above should help in customizing this model for individual use:

- ❑ Note 1: Invoice costs are the hard costs of the purchase, or the payments to the lessor. Both of these numbers begin with the lowest negotiated equipment cost possible. Because of the shorter term, however, **the lessor has a greater resale potential in the secondary market** (assuming the technology is current, generic, and in strong demand), and consequently can charge less money to the lessee. This results in the large cost savings (sometimes higher than this number, and often higher in sub-36 month scenarios) over purchase price.
- ❑ Note 2: Assets typically have to be entered into an asset database—independently of Fixed Assets software modules. Since most large refreshes use many different vendors' equipment (e.g., PC by one vendor, monitors by another, printers by yet another, software by multiple others), individual vendors will not be 'eager' to build and populate such a database with *other* vendors' data. This means the task of database creation/population/maintenance falls to the customer. But, a **vendor-independent lessor** has to do this *already* as part of the back-office function, and hence independent Lessors (i.e., vendors who have no allegiance to a particular brand) are often willing to share this with customers—as a complimentary value.
- ❑ Note 3: The lease management task is unique to the leasing scenario and represents any costs associated with managing invoices from the lessor. Since these are typically set up in an A/P module as recurring entries, this cost is fairly low. There are points where the cost is a little higher, especially at points where decisions about future leases and/or end-of-term processes occur.
- ❑ Note 4: This cost is the same for both lease and buy scenarios, since the actual labor required to transition user files, settings, and software from an old PC to a new PC is identical.
- ❑ Note 5: De-installation costs are likewise the same for both scenarios. This reflects end-of-life tasks, such as removal from networks (and access control systems), removal of corporate data and applications, and physical removal from user location.

- ❑ Note 6: Shrink/Warehousing is a cost unique to ownership scenarios. This represents the inventory shrink expenses associated with warehousing, equipment loss/theft/damage (plus a small amount of basic logistics expense, related to asset management). In the case of distributed assets such as PCs (mobile devices can be even worse), a firm would be fortunate indeed to 'lose' only 4-5 out of every 100 units, so this cost may be understated. We should point out, however, that 'shrink' is also reflected in leasing rates and programs, albeit in the rate structure. That is, some lease structures (such as CIT Flex Lease) are created *specifically* for less-than-100% returns (often in the 85-90% range), allowing a great deal of flexibility for the customer. Of course, the leasing rates are necessarily higher than full-return arrangements, but the financial advantages (to the lessor) of shorter-term leases means the 'pass through of shrink-loss' is also necessarily lower than in a full ownership scenario.
- ❑ Note 7: This is a shared cost, as well, reflecting simple shipping costs of a decommissioned unit to either the lessor or to a disposal facility.
- ❑ Note 8: Disposal costs are unique to the ownership scenario, since lessors bear this cost in the case of leasing. Disposal costs are rising steadily, so this percentage may already be too low. For example, one of the top 3 corporate desktop suppliers—with a vested interest in placing *no* 'barriers' to buying their new gear(!)—charges \$135 for disposal. On an \$800 PC, this is 16.8%. Some disposal costs are beginning to be charged up-front (under government mandates), to either the customer (e.g., California) or to the vendor (e.g., Maine)—who adds it into the selling cost, of course. Although some of these pre-paid fees are smaller, they are also *per item*—tripling the cost for a PC, a CRT, and a personal printer.

There are three alternative scenarios without professional disposal fees: reselling the used equipment (either through an independent wholesaler or directly by themselves) or donating the equipment to charity (for a tax benefit). **Reselling** through an independent **wholesaler** can *reduce* disposal costs to the upfront fees, and yield a positive 5% while staffing internally for a **direct** re-selling program is similar, but yields less—around 3.5%, due to internal staffing/tooling costs. These two scenarios are still significantly more expensive than the leasing alternative.

The **donate-to-charity** alternative has been widely used in the past, but has become less popular due to several problems (including the low value of the tax benefit at end-of-life). Many charities do not want the old PC's anymore since they often require as much as \$400 to bring up to reasonably current standards (Microsoft's estimate: they actually recommend donating to recyclers instead of charities, for some of these reasons), and some charities have become wary of firms 'dumping' broken gear on their shoulders (passing the disposal problem on to the charity). But, the largest problem is one that is growing: the data privacy risk. Getting the corporate data off a hard drive completely is an expensive process, and for firms with high data privacy requirements, the older *non-destructive* methods are no longer adequate—even when all legal liability is assumed by the charity. In any event, the tax benefit for a late-life PC is very small, comparable to the yields of remarketing.

Any complete Lease-versus-Buy analysis must take **all** of these costs into consideration—financial decisions based on only a subset of the data are inherently higher in risk than those which consider the full set.

Three Challenges to Harvesting These Cost Advantages

This increase in the rate of innovation offers the cost advantages noted above (i.e., more capacity for the same dollar, lower support/obsolescence costs for distributed assets, lower cash-out expense on faster-refresh leasing), when a faster-refresh policy is implemented. However, there are three potential challenges to actually realizing or harvesting any of the cost savings (or capacity enhancements) 'earned' by this approach.

One: Faster refresh cycles entail higher migration costs. That is, if I move from a 40 month refresh cycle to a 20 month refresh cycle, I obviously double the amount of install/de-install work my IT team has to do (or doubles the work my outsourcing partner will charge me for!). Since labor is often much more expensive than equipment costs, I could potentially 'use up' all of my savings on this additional labor cost.

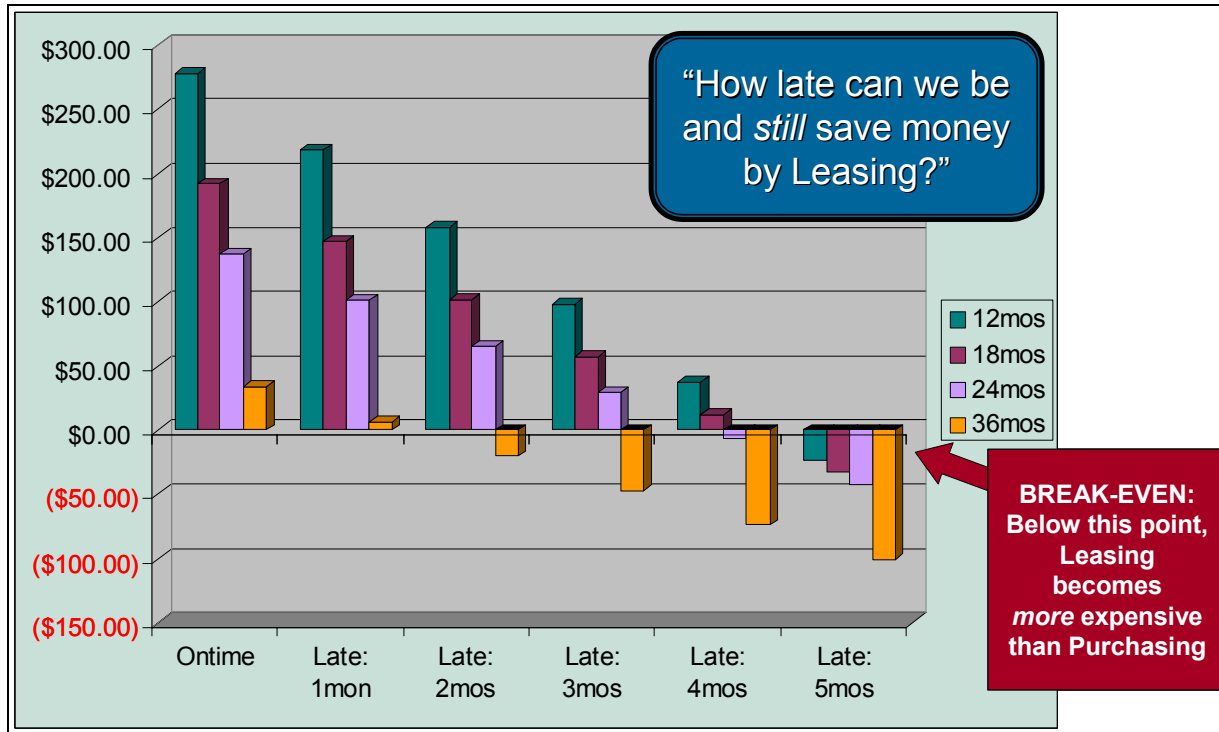
Evaluation: This challenge would have been a show-stopper even as recent as 10 years ago, but with the rise of automated imaging, provisioning, configuration and net-connection tools (e.g., in standard desktop management suites like Altiris or Microsoft's SMS/MOM family), the only remaining "per-PC" labor cost is the physical delivery and assemble step [the back-end reverse process is already accounted for in the TCO model above]. Gone are the hours/half-days at the user's desk, formatting drives, installing Windows and applications, and setting up customer-specific configuration files. And, this process is actually getting simpler, as our hardware vendors work hard to make migrating to their newer equipment 'easier'! So, while the number of installs is increasing, the cost associated with those installs would be less than the 20-40% upside in capacity increase.

Two: Some of the TCO-related savings are internal labor costs—employee hours spent on certain tasks—and thus represent savings that cannot be harvested (i.e., I still have to pay the employee for the hours, even if I don't have to consume those hours on distributed assets TCO-related costs). So, these savings are theoretical, and they assume that the costs are displaceable and/or disposable.

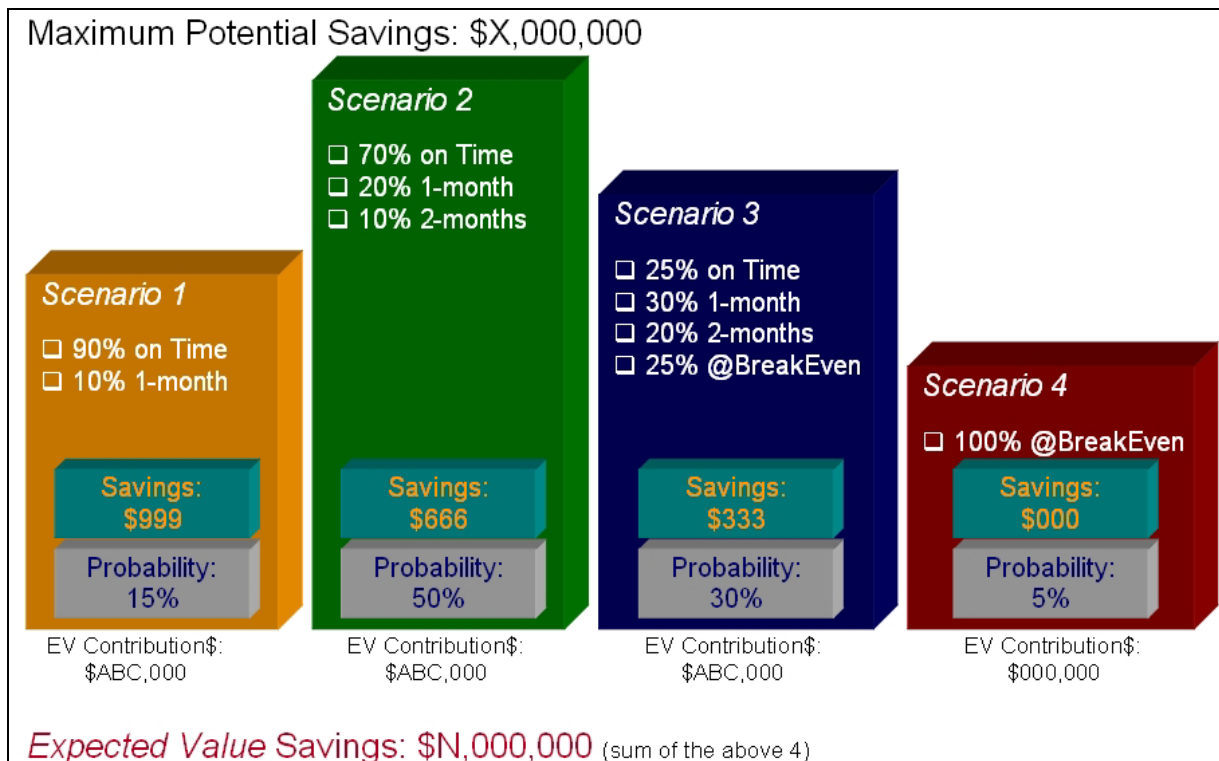
Evaluation: All non-specialist employee hours—at some critical mass—are displaceable, because work assignments can be consolidated and reassigned to fewer employees. If I have 8 people on my team, and I implement some new tool, training, or process which makes each of them 12.5% more efficient, then I can (theoretically) reduce my headcount to 7, if the work doesn't require skills not possessed by the remaining 7. We do this every day with reorganizations, headcount reductions/freezes, and efficiency initiatives. Sometimes we harvest the labor hours through reductions in force, but sometimes we instead use the freed-up hours/capacity to improve customer service, tackle new value-creating initiatives, or speed-up projects important to the business. This latter approach would not manifest cost savings, but *should be manifested in value creation metrics*. Harvesting these hours just takes management will and good planning—the same skills we use in reorganizations and workforce rationalization now. [This assumes that we are not outsourcing this labor, in a variable fee contract basis. If we *are* doing such, reduction in support calls will have a direct cost savings.] This challenge applies to *any improvement* initiative (e.g., training, new systems), of course, but in our specific case here we should note that (a) the support cost savings—based on the Intel study—are mostly labor support costs; (b) the cash-flow savings in the leasing scenario were pure invoice/hard costs; and (c) the administrative TCO-type savings were predominately non-labor in nature [only the asset database/lease management line items were labor-based and relevant; the rest were purely financial or a 'wash']. The 'hard' invoice savings (and cost-avoidance, in the case of the servers) are enough to justify the faster-refresh scenario, and by implication, the leasing of those assets.

Three: The savings in the leasing scenario presuppose adequate end-of-term asset management processes (e.g., returning the equipment on time, in good condition), and would evaporate quickly (into losses, even) by sub-optimal asset management practices. It does no good to compare Lease-versus-Buy scenarios on the basis of *unattainable* savings figures.

Evaluation: A sound financial comparison between Lease-versus-Buy scenarios should take this risk into account *from the start*. All that is required to derive a 'more sober' savings estimate is an Expected Value calculation. Since the shorter the refresh cycle, the greater the savings from leasing, decision makers should first calculate the 'break-even' point for their anticipated return time. Using the rates/costs on the following chart for illustration, we would first answer the question of "How late can we be and *still* save money by leasing?"



This is followed by constructing possible outcomes with different ‘mixes’ of returns (e.g., some on time, some 1 month late, some X months late), and then estimating some probability of each outcome. The Expected Value calculation is then straightforward: the sum of each outcome’s savings multiplied by that outcome’s probability. A diagram illustrating this shows the procedure:



This method will allow process risk to be factored into the calculation, yielding a much more 'comfortable' estimate of potential cost savings. This could also form the basis for incentive compensation for the asset manager: they could share in any additional savings achieved beyond the EV, through their direct enforcement of end-of-term directives. [It should also be noted that the leasing industry has adapted to these distributed asset realities as well, creating financial innovations such as CIT's Cost Certainty Leases (placing a guaranteed maximum on expense) and CIT's Flex Lease programs (with partial returns, relaxed return conditions, etc). This means that you would have a variety of structures to choose from—given the maturity of *your* back-end process—and could change the vehicle as those processes changed.]

Process Change: Implications and Suggestions

The discussion above suggests the following enhancements/changes to our Lease-versus-Buy decision process:

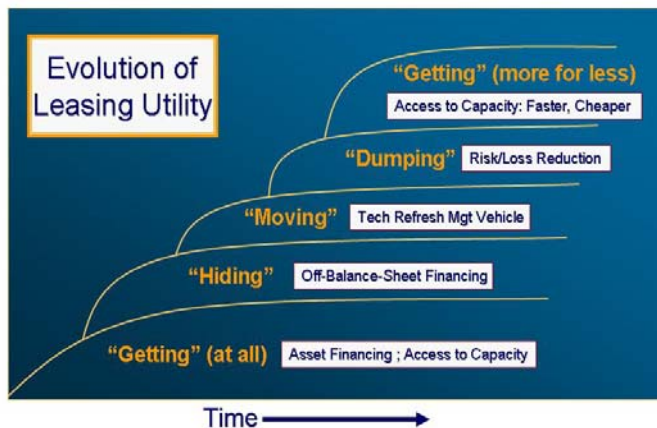
- ❑ Incorporate a refresh-cycle discussion/determination into our process **before** the actual Lease-versus-Buy models are populated. This discussion would need to include capacity projections for this specific class of equipment and obsolescence issues for distributed assets.
- ❑ Work with IT to tally support costs over the current refresh cycle and calculate the difference between the sum of years 1, 2, 3 and the sum of any years beyond that. **Include this 'extra' support expense** into the Lease-versus-Buy model, for any purchase/ownership scenario *longer than 36 months*.
- ❑ Assess the amount of entry-level and mid-level server spend (and how it is trending), to form an initial idea of what the potential same-dollar-higher-capacity and/or less-dollar-same-capacity opportunities might be. Look at each request for equipment in this class as an opportunity to leverage this. The group requesting the equipment should be required to submit a projection of capacity needs (for the business tasks to be run on this equipment), and **that projection**—once qualified—**should be used in planning the refresh cycle/leasing term** for *that* asset.
- ❑ Develop a refresh-cycle **stratification policy** for distributed assets. Some CIT customers already do this, having 12-18 month refresh cycles for certain classes of employees (e.g. software developers, sales reps, executives), slower cycles (24-36 months) for normal office workers, and 48-60 month cycles for fixed-function terminals (e.g., order entry, in a thin-client architecture). Similar to having different configurations of PCs for different classes of employees (e.g., more media/graphics options for Advertising/PR, less for Logistics/Transportation), this policy would facilitate a *best-fit* financial solution.
- ❑ Make sure realistic disposal costs are included in the calculation. Secure-and-compliant disposal of IT assets has become a specialty skill-set, and in-house groups which dispose of furniture, desk accessories, and paper documents are typically not equipped, trained, or given the expensive tools required to perform this task. Outside service firms are becoming more important for this work. This is a large expense percentage (and growing with the increasing government assessments of up-front e-waste fees), and one that occurs even in trickle-down scenarios (i.e., one employee gives their old PC to another employee who doesn't need a brand new one—this only postpones the disposal issue, not eliminates it).
- ❑ If there is some question about the efficiency of your back-end end-of-term asset handling processes, then build a realistic Expected-Value model (of end-of-term asset management processes) for 'discounting' possible savings from leasing, and apply this to *aggregate* lease projections (not necessarily *individual* cases). Continue to refine those end-of-term process to move this EV model upward, as you try to realize more of the potential savings offered in leasing arrangements.

- ❑ Run the Lease-versus-Buy calculation for *multiple* terms and then discuss the trade-offs with IT. The shorter the lease term, the greater the savings generally, but IT will bear the cost of the equipment turnover. Have them express their requirements and what they can reasonably commit to. In cases of central infrastructure equipment, however, they stand the most to gain from shorter cycles, since capacity pressures fall squarely upon their shoulders (i.e., when the machine is too slow, no one is blamed but IT!).

Implementing some/all of these refinements will help bring the important Lease-versus-Buy evaluation process more into line with technology industry trends, and allow you to harvest cost savings and/or capacity gains.

Evolution of Leasing: Adaptation to Faster-Refresh Cycles

Leasing practice has also been forced to adapt to this technology-intensive world. Customer needs for capacity, flexibility, and speed have created numerous innovations in IT leasing services. Distributed assets are notoriously difficult for organizations to manage (as assets), and steep obsolescence curves make the lessor's life 'interesting' too. In many areas, of course, the dynamics of leasing have remained the same for five thousand years—land leases from 3rd millennium BCE Mesopotamia are similar in content to modern leases (although easier to understand!), as are leases for boats and buildings. But, as the Industrial Age opened up new approaches to turning capacity into value, the need for access to productive capacity became acute. One financial vehicle that became popular was the capital lease, as a basic financing tool. It was not long, however, until more of the possibilities inherent in the concept of "other-owned assets" were exploited by savvy executives. Leasing developed new and different utility propositions, with a recent focus on **capacity management/planning**.



- **Access to Capacity** ("*getting*" more capacity, for less investment, in a capacity-critical environment)
- **Risk Mgt/Loss Containment** ("*dumping*" the assets when required, with a lower loss than buying; used for higher-risk projects/ventures)
- **Asset Management** ("*moving*" assets in/out of an organization, primarily for tech refreshes)
- **Financial Profile Mgt** ("*hiding*" the asset via off-balance sheet financing)
- **Financing** ("*getting*" the asset at all)

As you can see from the diagram above, technology leasing has grown into a multi-purpose tool, with some firms using it for simple financing, others for managing refreshes, and still others for risk/loss containment. In this paper, we have been focused only on the financial uses of leasing (i.e., access to capacity, expense reduction), but there are many other situations **in which leasing is used without any discussion of Lease-versus-Buy analysis** (e.g., getting test systems for only a year; requiring leasing for a high-risk business initiative—to provide a 'softer' way out in case it doesn't work).

Conclusion

In this paper we have explored the impact of faster technology innovation. We have seen that the rapid spikes in capacity provide an opportunity for today's enterprises to better match spend with capacity requirements. We have seen that shorter refresh cycles provide better access to capacity (for entry/mid-tier servers) and reduce/eliminate significant late-life obsolescence costs (for distributed assets). And we have seen that when leasing is applied to these faster refresh scenarios, the potential for 'hard cost' savings can be compelling. Of course, realization of some of these benefits requires management investment, but many of these savings can be easily harvested.

Faster refresh approaches will be implemented **selectively**, since not all areas of technology are experiencing the same rate of change as the ones discussed above. Some IT technologies will still be best analyzed using more conventional Lease-versus-Buy approaches.

As always, we at CIT Systems Leasing stand ready to work with you to design financial vehicles tailored to your organizational requirements and crafted to achieve the best possible mix of financial benefits, flexibility, and innovation. We can help you identify and harvest many of the benefits of selective fast-refresh initiatives.

The information contained herein has been obtained from sources believed to be reliable. CIT Systems Leasing disclaims all warranties as to the accuracy, completeness or adequacy of such information. CIT Systems Leasing shall have no liability for errors, omissions, or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice.