



A strategic comparison of Windows vs. Unix, version 2.0

Here's how, when and why a Unix/Linux-based architecture isn't just cheaper, but smarter too.
February 26, 2003, <http://linuxworld.sys-con.com/story/32639.htm>

Summary

Murphy's October 2001 TCO analysis generated much reader comment. In this article, the first installment in a two-part series, he revisits the Linux-versus-Windows decision for the faculty of a small college or university. (3,000 words)

By [Paul Murph](#)

(LinuxWorld) — This article compares the Microsoft client/server architecture to the Unix approach in terms of systems decisions facing a university faculty. To put this in context, imagine that you are being interviewed for a job as the faculty's systems manager. The chairman of the selection committee asks you to come in to discuss whether the faculty would be better served if it went all-Linux instead of staying all-Microsoft.

Notice that you're not being asked the traditional "which is cheaper?" question. Most people have little trouble figuring out that free is cheaper than not-free. What the committee members seek is something to help them reach a judgment about you and maybe some validation as to what they were thinking when they put you on their short list. They won't be evaluating your grasp of the facts. At this stage, they'll be looking for indicators of your behavior and attitudes. Are you a leader or a follower? Do you understand what drives the people and structures in their organization? Do you care?

Does your strategic vision of computing align with their needs? What kind of risks would they be taking if they voted to offer you the job?

Their needs & your deeds

First off, you might as well tell them up-front that you're a Unix evangelist. It isn't likely to be a secret, and there is always someone who'll chalk up a point or two for honesty. That said, you have to get past the initial ritual bowing to the reality of transitions and the risks of asking people to change behaviors. Getting the transition done is the hardest challenge you'll face if they offer you the job, but if you talk about it now you won't be *getting* the job. So mention the transition's importance and quickly move on to the good stuff: the clash of computing philosophies and your view that systems should be "invisible" — just magically there when needed.

To make sure everyone's on the same page, you should begin by defining your terms and your understanding of the uses for the system. For the faculty, those are as follows:

- operational access to systems managed by the university
- e-mail
- document management and preparation, including all the OpenOffice.org stuff
- student access
- educational programming (mainly Web-based course development and delivery)
- Internet services

Be aware — and *show* them that you're aware — that some faculty members will have personal favorite packages that they won't give up and can't be ported. Mention that there are many ways to accommodate these packages under desktop- and server-level Linux, right down to plugging in a Windows machine.



Point out that you're not going to compare Linux to Windows/XP Professional. Instead, you're going to be

Advertisement

Appro High-Performance Servers

AMD Opteron™ Processors

- 1U and 2U Cluster Solutions
- HyperBlade Cluster Solutions
- Simultaneous 32 & 64-bit computing



comparing systems built around the Microsoft client/server model to systems built around the Unix shared-access model.

Key differences

The Microsoft client/server architecture requires that the user interact with a Microsoft operating system running Microsoft applications on the latest Intel-powered PC. The machine, the client operating system and those client applications then interact with other Microsoft applications running on other Microsoft operating systems installed on PC "servers."



The Unix business architecture, in contrast, relies on smart displays to provide desktop access to centralized Unix servers. These, like the NCD shown above, typically offer large screens, high resolution, fast graphics and extreme reliability. They have no moving parts and no user-accessible operating system. People just turn them on, log in to the machine of their choice and start using applications.

And another thing...

The basic principle behind open source and the "release early, release often" strategy was described in 1965 by Multics designers [Corbat and Vyssotsky](#), when they wrote:

It is expected that the Multics system will be published when it is operating substantially... Such publication is desirable for two reasons: First, the system should withstand public scrutiny and criticism volunteered by interested readers; second, in an age of increasing complexity, it is an obligation to present and future system designers to make the inner operating system as lucid as possible so as to reveal the basic system issues.

The Internet allows more people to communicate more quickly over larger distances, but the principles haven't changed, and the evolution of Linux as a Unix implementation isn't structurally different from the evolution of BSD or Mach.

The most-fundamental differences between the two architectures have nothing to do with direct cost, but rather user behavior and the relationship between users and the systems group. The chairman mentioned Linux, but everyone needs to remember that Linux is Unix; the Linux kernel and the GNU utilities shared by Linux and Unix help implement Unix's standard set of ideas about user control and how computing should be done.

When Dennis Ritchie described part of the motivation behind his original work on Unix, he didn't say "the network is the computer," mention Linus Torvalds or talk about open source... largely because it was 1979, and a lot of stuff hadn't happened yet. What he did say in [The Evolution of the Unix Time-sharing System](#) set out the philosophy behind all of those:

What we wanted to preserve was not just a good environment in which to do programming, but a system around which a fellowship could form. We knew from experience that the essence of communal computing, as supplied by remote-access, time-shared machines, is not just to type programs into a terminal instead of a keypunch, but to encourage close communication.

That Unix orientation to sharing and communication is the exact philosophical opposite of Microsoft's proprietary stance. The PC community's "one man, one computer" mantra started out as a rallying cry against centralized control but became a lie as organizational needs for collaboration, control and security were re-asserted in PC environments.

Organizationally, PC use started out to be the equivalent of putting a brain in each leg of a centipede and expecting it to jog. To make this work, the organism had to evolve a central brain with the ability to ruthlessly suppress all the others. That's what has happened to corporate-PC use; desktop lockdowns and centralized servers mean that a lot of business-users are now discovering they've traded the reliability of IBM's 327X terminals for an unreliable GUI owned by an unresponsive help desk.

In a Unix-based architecture, lockdown makes no sense, as there's very little a user can do to affect overall systems operations. As a result, the user has real freedom to use a much larger resource and get help when needed.

Think about the students, too

In the case of a teaching faculty, a systems decision affects both its own costs and those incurred by, or on behalf of, its students. If the faculty stays all-Microsoft, individual students will have little choice but to go the same way. Otherwise, they could find themselves at a disadvantage relative to those who do.

If you looked at this issue from a purely technological perspective, you would say that students using products

such as Konqueror and OpenOffice.org on Linux at home would have little trouble working with the faculty's Microsoft systems. This is true; they can access most IE sites and convert Word and Excel documents back and forth quite easily. However, look a bit more closely and you'll see that relative to the faculty, the student's role isn't like that of an employee at a business or other organization. The work/home separation characteristic of the employment relationship doesn't exist here. Instead, the needs of the faculty impose themselves as a kind of distributed working environment on off-campus life.

In an employment situation, the power relationship between employer and employee ends when the employee goes home. Even if they are doing office work, employees using home computers are usually free to bypass Web sites that limit themselves to Windows clients. In most cases, they're also welcome to use office time and equipment to adapt documents prepared at home to Word's rather odd ideas about font-usage when converting them to the office-PC environment. But students don't have those options. They can't "just say no" to online exams that assume a Microsoft client or argue that Excel's solver is incorrect when some prof uses it to get the "right" answer on a small linear-programming problem assigned as homework.

There's an old saying about a chain being only as strong as its weakest link. The applicable variation of this adage is that the student's freedom of choice is constrained by the most proprietary piece of software the student has to accommodate. If the faculty picks Unix, the students will be able to pick almost anything for home use because there aren't any significant proprietary elements to worry about. If the faculty chooses to continue with the Microsoft PC, then the students, too, will be restricted to that.

As a result, PC costs for the student must be counted as consequences of the decision the faculty makes.

Looked at over five years, the direct costs for both architectures play out in terms of the initial outlay for hardware, software and set-up, plus the operating costs that go with user support and the kind of "evergreen maintenance" required to keep pace with internal and external change.

Given 50 or so faculty members, five non-systems support staff, and the need to maintain about 450 workstations for student use, mid-February 2003 costs for the Microsoft PC client/server architecture look about like this:

Hardware	Software	Unit Cost	Total Cost
At the School: Dell GX260, 17" flat screen, 256 MB of RAM, 40 GB hard disk, 2.8-GHZ P4; Canon bubble jet printer	Windows/XP Office/XP Pro	\$2,063 \$219	\$1,163,820 (510 units)
At Home: Dell Dim 8250, 17" monitor, 256 MB of RAM, 120 GB hard disk, 2.4-GHZ P4; Canon bubble jet printer	Windows/XP Office/XP Pro	\$1,458 \$349	\$650,520 (360 units)
Rack of four Dell Poweredge 2650; 2 x 2.4-GHz Xeon, 2 GB of RAM, 2 x 36 GB disk Four units: Dell Powervault 770N NAS 438GB	Windows 2000 back office servers Estimated as Small Bus. Server 475 CALs @232	\$38,944 \$42,100 \$110,200	\$191,244
36 Month Hardware Refresh			\$1,658,024
24 and 48 month software refreshes			\$695,060
Staffing and related, including overheads	5 full time	\$75,000	\$1,875,000
School Total			\$4,806,958
Parent Total			\$1,426,680
Five year direct TCO			\$6,233,638

Monitoring cost increases

One of the interesting things about this tabulation is that PC-costs have gone up quite considerably over the last two years. Those increases happened mainly in two areas:

1. The 17-inch, CRT-style monitors appropriate in October of 2001, when [an earlier version of this comparison](#) was published on *LinuxWorld.com*, have been replaced by LCD flatscreens.

Flatscreens aren't just popular in the Windows-PC world — they're valuable, too. Microsoft's ClearType technology, available with XP and mostly applicable to LCD displays, offers clear advantages in readability. These advantages almost certainly correlate directly with the student's ability to understand and remember material presented on screen. Thus, these monitors are as much a requirement for PCs used in education as ABS brakes are on a new car for use in Canada or the northern-tier states.

2. This configuration uses a more-expensive storage area network with its own memory and Xeon processors instead of the SCSI storage array appropriate 15 months ago.

Storage area networks started out as a solution to the storage-management needs associated with unconstrained Microsoft-server sprawl, but Microsoft's licensing demands and software adaptations to use of storage area networks are rapidly making them mandatory components of professional practice in any

Microsoft-server environment.

In contrast, the faculty IT group is significantly underbudgeted, with only five people — one of them a manager — to support 500 users. The Dell pricing therefore includes three years of next-day, on-site support for the gear exposed to student use, and it is implicitly assumed that most of the faculty will act as their own front-line support for themselves and their students.

On the Unix side, however, costs have continued their decline, allowing this comparison to be based on 21-inch NCD smart displays for the faculty and 19-inch home-PC screens supporting Linux (instead of the 17-inch screens specified in [the previous analysis](#)).

For the faculty, the Unix direct-cost advantage starts out at about 25 percent of the cost of the Windows architecture. That number improves to an estimated 50-percent savings after five years.

For the parents, however, the range is wider and the pattern is different. Their initial capital outlay is only trivially smaller for Unix than Windows — largely because they buy higher quality gear — but exceeds 50 percent after five years because their students eventually leave home still using those original Linux PCs.

Hardware	Software	Unit Cost	Total Cost
505 x 21" NCD smart displays	NCDWare	\$1,620	\$818,100
2 x Sun V1280, 12 x 900-MHZ CPU, 1.7TB Disk (2 x 876 GB), 32 GB RAM	Solaris with all needed application software	\$219,900	\$453,790
Administration workstation: SPARCstation 2000; 2 x 1-GHZ CPU, 4GB, 2 x 73 GB disk, CD-ROM & floppy drives, 21-inch monitor	Solaris	\$31,254	\$31,254
At Home: Dell Optiplex GX260 19-inch, 256 MB, 2.8-GHZ P4, Canon F60 Printer, with SCO/Linux	OpenOffice and related open source tools	\$1,760	\$633,600
Staffing: includes overheads	1	\$120,000	\$600,000
Cost to College	Includes 20% annual maintenance		\$2,162,300
Cost to Parents			\$633,600
Start-up total			\$2,798,880

There are several things to notice here:

You couldn't make this stuff up

During the early days of the GUI, a lot of serious researchers worried that reading from a CRT was harder than reading from paper, resulted in reduced comprehension and lowered information-retention.

That argument was more-or-less settled in 1989-90 through work done by Mutter et al at the University of Toronto. Their results (see *Behavior & Information Technology*, VOL. 10, NO. 4, 1991, 257-266) showed that reading from paper and CRT displays produced roughly equivalent results.

Their work was done using Macs with Radius full-page displays, but it was widely misinterpreted as applicable to PCs running Windows.

In those days, Apple made its own graphics cards, and the Macintosh used in the experiment benefited from something very like XP's ClearType — a technology Apple first applied on the 1979 Apple II Plus.

1. This configuration is based on using two V1280s to provide redundancy and a total of 24 CPUs accessing just over 3.5-TB of UltraSCSI disk.
2. Physical networking costs have been ignored for both configurations. In reality, the Unix approach avoids all of the Windows vulnerabilities while offering significant savings on hardware and maintenance.

To return to the job-interview metaphor, you could sensibly say that one of your first jobs after the system reaches operational stability will be to explore having the faculty set up its own DSLAM to offer local ADSL connection services to students and faculty. That, of course, requires policy endorsement by the faculty and university. But if it's viable and approved, it could lead to full cost-recovery on network operations while offering both faculty and students cost-reductions on home-access.

3. The comparisons show five full-time support people for the Windows architecture but only one for the Unix option.

This is far too low on the Windows side, with most researchers suggesting that total Windows support staffing should run in the 40:1 range, not the 100:1 range shown here. For example, a recent [Booz-Allen &](#)

[Hamilton study](#) for the Virginia Department of Education, for example, shows more than 11 FTE (Full time Equivalents) deployed to support operations on only 371 Windows computers and 59 Macintoshes for a ratio of 39:1 (note that some "studies" do show ratios of 100:1 or better, but if you read those carefully, you'll see that the number usually applies only to help-desk staffing and not to the entire support operation).

On the other hand, the Unix number may seem low. You do sometimes see "studies" that show Windows support and maintenance costs to be lower than those for Unix. Generally speaking, what those demonstrate is that if you simply select the right sources, you can get any result you want to pay for. Interview people who use Unix as a cheaper mainframe or Windows server, and their staffing costs will be higher than those cited in a pro-Windows ad or white paper. Interview people whose Unix-use consists of a five-year-old HP-UX machine running Oracle Financials for 1,000 users, and you can compare their results to those from people running personal Linux Web servers to discover that Unix can be 80 percent cheaper than Unix. This kind of thing depends on what questions you ask and, more importantly, what you don't ask.

In the real world, however, Unix is kind of fire-and-forget. Get this system working right, fire anyone who logs in as root without sufficient skills and a well-thought-out reason for doing so, and it'll run by itself for years.

You might have noticed the exorbitantly expensive workstation I gave myself as a signing bonus here. Yes, it's a toy and it's overkill, but there's also a serious purpose. Every change, every piece of new software on one of the operational machines will be run on the workstation first. That's because the overwhelming majority of Unix-system failures arise from administrator action. Using this workstation, I could pilot changes, control implementation and eliminate almost all of those failures, avoiding "fat finger syndrome" in the process.

4. There isn't any need to replace the Unix infrastructure as part of the evergreen process. Microsoft's product-churn doesn't happen in Unix; Unix software gets *better* — not *fatter* — over time. Once Unix systems work, they'll continue working. Software upgrades will actually be upgrades, not hardware requisitions.

Think about that for a moment: there are tens of thousands of ten-year-old Unix machines competently running current software with 24 x 7 reliability, but you won't find any ten-year-old 486s running Windows 2000 Server with current application-loads. Most people who bought Sun or HP servers five years ago still use them, but people who bought NT servers with Windows 95 desktops have typically had to upgrade their hardware twice and their software three times, just to be ready for the next round of upgrades this year.

The intangibles

There's also a key organizational difference that doesn't show up in the cost table. The Unix system provides near-perfect reliability and freedom from both student attacks and external attacks on system integrity. Remember Slammer? Lirva? Code Red? Whatever today's special Windows horror is? Most of this stuff just doesn't affect Unix operations, although Microsoft boxes on the Internet can pollute the network to the point of slowing down external access.

It's easy to create cost-impact estimates for these differences. Using the numbers from bugtoaster.com, you can predict that the faculty would experience around 9,800 individual systems failures per year with the Microsoft client/server architecture... as compared to maybe *one or two* with Unix. Combine that with some assumption about file losses and time-to-remediate, and you get a big number. That's perfectly valid and clearly part of the cost of choosing Windows, but it's trivial compared to the impact that expected systems failures have on a user's behavior.

In the Microsoft client/server model, faculty members become part-time PC-support workers, continually interrupting themselves to deal with the latest crisis. This gradually reduces their own view of computing to that of the Windows PC while blanking out their awareness of other options. In many places, this is what we have now. Not only is it wasting a significant percentage of teaching resources, it's producing a generation of graduates that thinks SAP costs \$495 and runs on a PC.

In the Unix model, the computers work. They blend into the background like telephones and power plugs, letting teachers teach and researchers research.

As a result, the direct-cost comparison shows a Unix advantage in the range of 50 percent over five years, but the unquantifiable indirect effects are clearly much more significant. These costs, measured in terms of how well the faculty does its job, play out over the lifetime of the university's graduates and the careers of its teachers.

Stay with Microsoft and the need to work with the PC will gradually narrow your view of the computing world until all you can see — and all you can teach — is the hope that the next generation of Microsoft products will magically be effective. Go all-Unix and the computing infrastructure disappears from day-to-day visibility, leaving teachers

free to teach their subjects and students free to learn.

About the author

Paul Murphy wrote and published 'The Unix Guide to Defenestration'. Murphy is a 20-year veteran of the IT consulting industry.

Related Stories

- [Readers react to 'Weighing the pros & cons of IBM's mainframe Linux'](#)
- [Weighing the pros & cons of IBM's mainframe Linux, Part 2](#)
- [How LinuxWorld got Linux-on-a-mainframe wrong](#)
- [The pros and cons of business-app implementation via open-source software \(Part 1\)](#)
- [Open-source vs. Microsoft tools for business-app implementation, Part 2](#)
- [Seeing through the Linux-Windows TCO comparisons](#)
- [A strategic comparison of Windows vs. Unix](#)
- [Readers react to 'A strategic comparison of Windows vs. Unix'](#)

Related Sites

- [Booz-Allen & Hamilton TCO study for the Virginia Department of E](#)
- [Introduction and Overview of the Multics System](#)
- [The Evolution of the Unix Time-sharing System](#)
- [The Unix Guide to Defenestration](#)
- [Weighing the pros & cons of IBM's mainframe Linux](#)
- [Weighing the pros & cons of IBM's mainframe Linux, Part 3](#)

[Copyright © 1998-2003 LinuxWorld.com](#), an IDG Communications company.