Research Brief

UNIX/Linux and Windows - the Power of an Integrated, All-in-One, Familiar Environment

Introduction

A recent IBM information technology (IT) market study shows that labor now accounts for seventy percent of an enterprise's IT budget. And almost three-quarters of this labor bill goes to people-related costs for managers of systems, storage, networks, databases, applications, infrastructure, et al. With salaries that typically range from seven-five thousand dollars to over one-hundred and twenty-five thousand dollars per year for skilled specialists, it is easy to see why most midsized and large enterprises are interested in any technologies that can help reduce the cost of labor.

One way to attack the cost of labor is to find ways to automate rote tasks. For instance, if an enterprise conducts a systems back-up in much the same way, day-after-day, this rote task is a candidate for automation.

Another way to reduce labor costs is to reduce management complexity. For instance, an enterprise can standardize on a single operating environment and systems platform — ensuring that all devices in an information systems environment look and behave the same. Standardization reduces complexity, which in turn reduces management costs.

The reality, however, in most of today's more complex information systems designs is that midsized and large enterprises rarely deploy homogeneous information systems environments. More often than not, UNIX systems are used to run business applications and provide decision support; Linux is used to provide application and database services; and, Windows systems are used for office functions (mail, collaboration, personal productivity, etc.). So, how can enterprises reduce management complexity and improve interoperability across heterogeneous systems?

One way to mask heterogeneous systems management complexity is to find a Unix/Linux/Windows toolkit that can help an enterprise build a common management environment where systems can be automated and managed in exactly the same way by the same group of people. To do this, an enterprise must choose a common desktop (Windows, of course), as well as a toolset that can manage across Windows and Unix.

Only a few vendors offer these kind of cross-platform toolkits (including Microsoft, MKS Inc, and Quest Software). This report discusses where, how, and why such tools should be used — and provides comparative information on the above mentioned vendors. Microsoft's Subsystem for Unix-based applications (SUA) does not appear to be ideally suited for administration tasks (but could remotely administrate Unix). Quest has a different focus and may be useful to some. But the MKS Toolkit allows administration of Unix from scripts that run remotely on Unix or locally on Windows — AND administer Windows using common tools. Plus MKS facilitates using the same scripting expertise for both platforms.

Further, these Unix/Linux/Windows toolkits can be used to build scripts that can operate across platforms to automate business process flows across heterogeneous systems. And they can be used as "one-off" tools for building applications that run on Unix, Linux, and Windows.

Why Use Integrated, Cross-Platform Interoperability Tools?

There are several business reasons why an enterprise might focus on integrating (or bridging) UNIX and Windows environments. But the top three business benefits that can be derived by so doing are:

1. The ability to reduce management costs;

2. The ability to automate redundant tasks; and,

3. To smooth business process flows in order to create operational efficiencies.

To reduce management costs, enterprises must reduce management complexity. This can be done by using a common, consistent scheme to manage both environments. Less complex systems are easier to manage — and often require less skilled (and therefore less costly) individuals to perform management functions.

By automating redundant tasks, systems take on labor responsibilities previously performed manually. The more tasks that can be automated, the more money an enterprise can save.

Finally, applications and data often reside on disparate UNIX, Linux, and Windows systems. Enterprises that can tie these systems together using cross-platform scripts that enable applications and databases to work together harmoniously can more easily flow information across their information systems than those that don't. Cross-platform process flow enables greater business process flow efficiency which, in turn, results in greater profitability and increased competitiveness.

Where to Find UNIX/Windows Integration Tools

Building a cross-platform management and automation environment can be quite complex. System calls and commands between operating environments differ; each operating environment uses different application program interfaces and libraries; and so on… To manage across environments a common set of utilities, tools, and libraries needs to be built.

Fortunately, the vendor and open source communities have already built such a large set of tools, utilities, programs, and libraries.   Application developers, infrastructure specialists, and operating environment specialists can approach UNIX/Linux and Windows integration using a variety of free, open source tools and utilities — or they can buy tightly-packaged, complete migration/integration environments from a small number of independent software vendors (ISVs). Each approach has its merits and demerits.

The Open Source Approach

For those developers who choose to use an open source approach, the primary benefit is that most, if not all the code to be used for UNIX/Windows porting or integration "free". But, in reality, the only thing that is really free is access to the source code. Making the source code work in each unique environment requires varying degrees of effort (and in the long run can end up costing more than a commercially available, integrated, ISV solution).

The downside of the open source approach is, of course, that the onus is on the developer(s) using a collection of open source products to integrate the various open source components, tools, utilities, and libraries they decide to use into a cohesive UNIX/Windows management environment. And as developers attempt to integrate these components, some immediate questions come up such as: "how do you establish a standard set or even which versions of these commands and utilities to use?"

The other major downside for organizations who choose the open source approach is the lack of support. With open source code the concept of "24/7 support" now has a whole new meaning (the onus is now on the developers to be available 24/7 to fix the problems they encounter not only with the software they are building but also the tools they are using to build those applications). Further, by spending time supporting a tool set and a cobbled-together management environment, developers often find that this takes away valuable time from the jobs they were hired to do (and, accordingly, companies run the risk of project delays and setbacks).

There are several sources of supply for open source UNIX/Windows tools, utilities, and libraries. Some of these sources include the Cygwin project, and Interop Systems community.

> **The big drawback in using the open source approach is the amount of integration work that must be performed by developers in order to build UNIX/Windows integrated environments. The installation and deployment of the myriad of tools, utilities, libraries, and applications can be quite complex (read some of the open source forums where developers are requesting help to get a sense of how complex this can be). ISV UNIX/Linux/Windows integration products tend to be less complex to deploy — and certainly better integrated.**

ISV Environments

Surprisingly, there are relatively few ISV environments that provide UNIX/Linux/-Windows integrated application program/management products. However, Microsoft, MKS Inc. and Quest Software are three of the leading suppliers of such products. (We discuss these vendors' product offerings in greater detail in the following section).

A Closer Look at These ISV UNIX-to-Windows Integration/Interoperability Products

Microsoft offers two UNIX/Windows integration products: Services for UNIX (SFU) and Subsystem for UNIX-based Applications (SUA). Quest Software offers a variety of plug-ins that help UNIX boxes come under the control of Windows resources (such as Active Directory, Exchange, SharePoint, et al). And MKS offers its MKS Toolkit for Developers — a toolkit and related utilities and libraries that can be used to bring Windows resources under the control of the UNIX operating environment. This section looks at Microsoft's SUA and the MKS Toolkit in greater detail.

Microsoft SUA

Until recently, Microsoft's SFU has been the company's primary UNIX/Windows integration offering. But only a few months ago, Microsoft changed its strategy — discontinuing any further revision work on SFU and announcing that this product was going into an end-of-life cycle (terminating in 2011 with additional support available until 2014). Microsoft's new strategic direction for UNIX/Windows integration is SUA.

UNIX/Linux and Windows: The Power of an Integrated, All-in-On, Familiar Environment

Microsoft defines SUA as a "source-compatibility subsystem for compiling and running custom UNIX-based applications on a computer running a Windows server-class operating system". This subsystem is installed separately from Windows and operates as a guest UNIX operating environment on Windows systems (more precisely, it operates as a Portable Operating System Interface (POSIX) UNIX environment).

To support UNIX/Windows integration, SUA users can download UNIX utility packages from Microsoft's Web site (such as shells and a Telnet client). These utility packages include scripting utilities and a software development kit (SDK) to assist in UNIX application development.

> When it came to integrating UNIX and Windows, we noticed developers needed to turn to Microsoft and the open source community for downloads of several utilities, libraries, and APIs not supported by SUA. To us, this signals that SUA needs a more comprehensive feature set, better packaging, and better integration.
>
> Further, we could not locate SUA for Windows XP — and we observe that SFU does not run on x64 or Itanium Windows environments. In other words, Microsoft's SUA/SFU products do not appear to work across their entire product lines.

MKS Toolkit for Developers

In contrast to Microsoft's SUA, the MKS Toolkit for Developers product is a well packaged, feature rich environment with a comprehensive UNIX/Linux and Windows integration management scheme designed to manage Windows environments using UNIX commands and scripts. This toolkit features the abilities to:

- Unify and automate software builds across UNIX and Windows environments;

- Automate management tasks to increase productivity;

- Connect to remote UNIX and Windows systems (to perform troubleshooting or maintenance); and,

- Remotely administer Windows environments (update user and group accounts, permissions, security, etc.).

From a management perspective, the MKS Toolkit for Developers contains hundreds of authentic UNIX utilities such as *grep*, *df*, *du* and *ls*, as well as *ksh*, *csh* and *bash* shells (enabling UNIX developers to immediately start using familiar scripts to manage UNIX as well as Windows environments). Password synchronization, remote utilities, and daemons are also supported. And UNIX commands can be used to perform automated back-ups across UNIX and Windows systems.

> But where the MKS Toolkit for Developers really shines is in scripting between Unix and Windows environments. MKS offers tools like "ls", "mv" and "tar" that have familiar command lines for Unix-savvy developers — but under the hood these tools are deeply integrated with the Windows operating environment and file system. For instance, "tar" will cache access control lists (ACLs) and not only store Unix file permissions but also Windows file permissions. When a developer goes to extract these files, they can be automatically relocated and restored (when the tarball is extracted). In the Unix/Windows world, its hard to find products with this depth of integration — but the MKS Toolkit is one such product.

Futher, the MKS offering supports tools specifically for the Windows environment, such as service, shortcut, db and registry, which can be wrapped in scripts and chained with these traditional Unix commands to perform tasks that are specific to Windows environments.

Some of the other tasks that the MKS Toolkit allows users to perform include:

- implement automated, cross-platform software build scripts;

- use automated scripts to manage large Windows installations by populating and managing registry entries en masse;

- manage user accounts in batch mode (as opposed to one-by-one entries);

- create standardized scripts to perform automated management functions; and,

- use scripts to automatically perform tasks that may be somewhat difficult in the Windows world such as adding users and groups to a domain, performing complex file searches, performing UNIX-compatible back-ups, moving data and files easily between machines, clone a system file tree or document tree, and copy a standard user setting to a new user account.

Finally, we observe that the MKS Toolkit is also deeper than Microsoft's SUA — offering support for 500 UNIX and Windows command-line utilities, as well as providing very broad API support. And the toolkit's Visual SFTP Explorer extension allows drag-and-drop capability between UNIX and Windows.

We would be hard-pressed to argue that SUA is anywhere close to MKS Inc in terms of depth, and out-of-the-box integration.

Skills Bias

When comparing Microsoft SFU/SUA and Quest Software's suite of interoperability products to MKS Toolkit for Developers, we did note one major difference — a difference that we call a "skills bias".

Microsoft and Quest both lean toward a view of having Windows control UNIX environments. Indeed, Microsoft's new strategic direction with SUA is all about running UNIX applications under the control of Windows. And Quest offers products that enable UNIX boxes to fall under the control of Microsoft resources such as Active Directory, Exchange, SharePoint, et al. On the other hand, MKS leans toward using Unix-like tools used to manage the Windows environment and remote access to Unix to administer Unix platforms.

This difference in outlook really comes to a head when examining how scripts can be used to integrate UNIX and Windows environments. Scripting languages and familiar tools ensure that the valuable programming expertise of UNIX developers, and the power of commands such as scp, sftp, secure X, KornShell, CShell, awk, telnet server, xterm, rsh, rlogin, secure utilities and Perl, do not have to be discarded when Windows systems are introduced into a UNIX environment. Further, the use of these utilities to control Windows environments can also reduce the immediate need for cross-platform training.

On the flip side, companies like Quest and Microsoft take a view that Windows is as capable and powerful as UNIX — and that their approaches to manage UNIX as part of a Windows domain is equally powerful in managing cross-platform environments.

> **From our perspective, a major decision factor is a particular customer's "skills bias". If a customer is a big believer in Microsoft or Quest management tools and utilities, and if that customer wants Windows to be the driving integration force, then Quest and Microsoft become likely choices as the focal points for cross-platform interoperability. If a customer is a big believer in UNIX superiority in cross-platform management (and is a big believer in the power of UNIX scripting languages and automation capabilities), then the MKS Toolkit is the logical choice.**

Summary Observations

The name of the game when it comes to managing heterogeneous UNIX and Windows environments is to find ways to reduce management, administration, and support costs. One way to reduce these costs is to use a single group of support personnel to manage both environments (leveraging one skill set to manage both). Another way to reduce costs is to automate as much manual functionality as possible. By reducing complexity — and by reducing the amount of manual labor an enterprise must exercise to manage disparate systems types — enterprises can lower their cost of operations and significantly cut labor costs.

Several vendors — as well as the open source community — offer products, utilities, and tools that can help enterprises manage combined UNIX/Windows environments. The primary difference between the Microsoft/Quest Software approach and that of MKS is that Microsoft and Quest Software attempt to manage UNIX from Windows whereas MKS Toolkit products focus on using Unix-like tools used to manage the Windows environment and remote access to Unix to administer Unix platforms.

Both approaches (UNIX managing Windows, or Windows managing UNIX) could work — but we noticed significant differences in the depth, maturity, and comprehensiveness of the MKS Inc offering. The MKS Toolkit for Developers is a feature rich and very well integrated set of tools that are well-suited for cross-platform scripting and administration.

In the end, both approaches work and can help lower cross-platform management costs. The decision factors on which approach to use should be based upon:

- The skill set bias an enterprise has (do you want Windows or UNIX people managing cross platform integration);

- The level of integration work your organization is willing to fund and support;

- The amount of support your vendor or community is willing to provide your organization as it deploys its UNIX/Windows interoperability solution; and,

- Are you willing to pay an ISV for an integrated product — or do you want to take your chances by building a UNIX/Window interoperability environment yourself.