Wabi Backgrounder

Wabi™, a new technology from SunSelect, a Sun Microsystems, Inc. business, provides a new and effective solution for integrating PC and UNIX® computer resources. With Wabi, users can run off-the-shelf Windows applications on UNIX workstations without MS-DOS® or Microsoft® Windows and without altering applications for the UNIX environment.

Wabi will be an integral component in helping businesses make the transition from PC-based information systems to more powerful UNIX-based systems. As businesses make the transition, they will have specific requirements for integrating PC applications into UNIX systems. They will also need to know what tools are available to ease the transition, keeping users productive while they learn the benefits of the new platform. The following describes the issues in integrating PC and UNIX technologies and explains how Wabi can provide businesses with the best of both worlds.

Moving PC Users to More Powerful Platforms

Making the transition from PCs to open standards-based UNIX systems is not a simple matter of unplugging one machine and installing another. Companies want to maintain their investment in PC applications and leverage enduser training and familiarity with PC-based technology. Users want to continue to access widely available personal productivity applications for the PC platform.

Businesses making this transition face a dilemma -- how to easily integrate existing productivity applications, such as spreadsheets and word processing software, with the new, high performance UNIX platform. Some have solved this problem by maintaining two platforms on the same desktop, a PC and a UNIX workstation or terminal. Others, in an effort to consolidate the two environments into a single system, have explored the emulation of PC operating systems and applications on powerful UNIX systems. But neither approach has been fully satisfactory.

Requirements for PC Applications on UNIX

As users look to better integrate PC productivity applications with UNIX platforms, they want to be sure several requirements can be met. These include:

- Performance: PC users expect their applications to perform as well on UNIX as they do on native PC platforms.
- Compatibility: PC application technology for the UNIX platform should provide compatibility with both commercially-available PC applications and those written by in-house programmers.
- Integration: PC applications on UNIX should leverage the advantages of the UNIX environment. Chief among this is the ability to access UNIX network resources and UNIX multitasking capabilities.
- Investment: PC applications should be as seamless as possible with the native UNIX environment. Little else should be required to achieve desired performance and functionality.

Traditional Solutions for PC Applications on UNIX

A number of solutions for running PC applications on UNIX are currently available. None of these solutions, though, can meet all of requirements previously listed. While all approaches offer advantages to both the user and the application developer, there can be significant trade-offs in terms of cost, simplicity and performance.

Emulation Technologies

The most common approach to running PC applications on UNIX systems is emulation technology. PC emulation uses software, or a combination of software and hardware, to simulate PC hardware and system software as a UNIX process. PC emulation requires a physical copy of the MS-DOS operating system be present on the UNIX machine. Microsoft Windows must also be present if the user wants to run Windows applications.

To run a PC application on UNIX, the user boots the copy of MS-DOS, and Windows if necessary, and then launches the application. The DOS processes think they are on a PC because all of the PC hardware registers are active due to PC emulation.

Because PC emulation requires MS-DOS and Microsoft Windows to work, it provides a high degree of compatibility with applications written to those operating environments. PC emulation also allows MS-DOS applications to take advantage of the underlying UNIX operating system. By using the multitasking feature of UNIX, PC emulation enables the user to run multiple DOS sessions, the equivalent of having multiple PCs on one desk all operating simultaneously.

PC emulation can provide excellent performance levels for MS-DOS applications. However, the emulation process can tax native CPU performance. Additionally, users running Microsoft Windows-based applications need to add an extra layer of software to the process. This extra layer, the Microsoft Windows user interface, can slow application performance. New technologies are emerging to address performance issues involved in running the latest generation of PC applications on UNIX.

Software Developer Kits

Software Developer Kits (SDKs), a more recent development, can eliminate the need for PC emulation. With an SDK, application developers can recompile existing PC application source code into a format that can run natively on UNIX. SDKs recompile source code with new libraries which translate the Application Programming Interface (API) calls to the native system calls. For example, once a Microsoft Windows application is recompiled, the Microsoft Windows APIs are translated by the supplied SDK libraries to Xlib and UNIX system calls.

SDKs allow application developers to create an application that runs natively on UNIX but still has the look and feel of the original PC application. Once recompiled with the SDK, the PC application can have higher performance levels than those running under PC emulation. Users can also enjoy native UNIX system capabilities, including the mouse, keyboard and video.

SDKs require developers to recompile application source code for every desired operating system and subsequent revisions of those operating systems. As a result, multiple versions of the same application have to be created and supported. Recompilation technology has not yet been perfected and not all APIs can be successfully translated to new libraries through SDKs. Developers using SDKs may have to invest significant time and money writing new source code for the ported application or compromise on the functionality of the recompiled application.

A New Approach to Windows Applications on UNIX - Wabi

Wabi provides a unique approach to running PC applications on UNIX. Unlike PC emulation technology, Wabi does not require MS-DOS or Microsoft Windows to work. Nor does Wabi require Windows applications be recompiled for the UNIX environment. Wabi allows Windows applications to run "off-the-shelf" on UNIX by translating the application to the UNIX environment while the application is running. As a result, Wabi provides increased Windows application performance and requires lower system overhead.

On the average, Windows applications written to the Win16 API spend most of their time making calls to the Windows kernel (about 60% to 80% of all calls). Most of these API calls are associated with the User Interface aspects of the application, such as menus, displaying text, and scrolling. Wabi translates the Win16 API calls to Xlib calls, the set of APIs for X Windows environments based on the X11 standard.

Wabi allows Microsoft Windows applications to run on both RISC- and X86-based UNIX systems equally well. When running Wabi on an X86 version of UNIX, instructions that aren't handled by the Windows API are passed directly to the X86 processor, just as they would be in a native Microsoft Windows environment. In RISC implementations of UNIX, Wabi uses a CPU simulator. The simulator translates X86 instructions to the associated RISC instructions, executes them and returns the appropriate value to the Windows application.

Benefits of Running Windows Applications on Wabi

Because Wabi imposes lower overhead on the UNIX operating system than PC emulation technology, Windows applications can perform as well as or better than they do in a native Microsoft Windows environment. The benefits of Wabi go beyond just performance issues, however.

Based on the Win16 APIs, Wabi provides a high degree of compatibility with all major Windows 3.X applications. Wabi preserves the functionality of these applications, supporting OLE/DDE links between applications and allowing users to drag and drop between applications. Wabi also eliminates the complexity caused by SDKs. Microsoft Windows application developers can write to one set of APIs and maintain one version of the source code, knowing it will run on both Microsoft Windows platforms and UNIX platforms with Wabi. TrueType font compatibility, important to Windows application users, is provided in Wabi by using font handling technology from Bitstream, Inc.

Because Wabi translates Microsoft Windows applications to the native UNIX desktop, endusers have the added advantage offered by the 32-bit UNIX operating system and underlying hardware. This includes:

- Support for multiple, simultaneous Windows applications at native (X86-UNIX) to near native (RISC-UNIX) performance
- Native system video, mouse and keyboard performance
- Ability to cut and paste text and X graphics between UNIX and Microsoft Windows applications
- Supports OLE/DDE between Windows applications
- Support for native X fonts and TrueType fonts
- Access to UNIX network resources, including files and printers, from within Microsoft Windows applications
- Access to UNIX file system, serial ports, and network printers
- Support for multiple users via X-terminals
- Supports Windows enhanced (386) mode applications
- Reads and Write DOS diskettes
- Fully scalable windows and fonts

Wabi Provides the "Best of Both Worlds"

As companies transition to powerful UNIX platforms to run their businesses, compatibility with existing PC-based personal productivity applications becomes an important issue. Users want the benefit of having PC and UNIX applications on one platform without increased complexity or performance sacrifice. Application developers want to expand their market opportunity without supporting multiple versions of source code and writing to multiple APIs. Systems administrators want to provide PC and UNIX compatibility without supporting multiple platforms and increasing system overhead or investment. Wabi provides the best of both worlds for endusers, developers and MIS managers. With Wabi, UNIX-based systems provide compatibility with existing PC applications while offering a single, high performance platform.