MIGRATION SPECIALTIES

217 West 2nd Street, Florence, CO 81226-1403
+1 719-371-1711, Fax: +1 888-854-3417
Info@MigrationSpecialties.com
www.MigrationSpecialties.com

Bruce Claremont, Updated: June, 2009

Emulating Critical Legacy Systems

As organizations consolidate IT operations, there always seems to be a few critical systems running on legacy platforms that prove difficult to replace. These systems are crucial to operations, are stable and reliable, and provide functionality difficult or costly to replicate in the new environment. Often these systems have reliably run in the background so long that the only reason they are being noticed is a) they do not fit the new platform consolidation mandate; or b) the hardware upon which they operate has begun to fail.

In some cases, the applications running on these older systems have modern counterparts. Moving to the new application is generally costly and often painful, but at least a migration path exists. However, some applications are truly unique. Either hardware components are highly specialized and no longer produced or the software is unique and no longer available. Process control systems, automated test systems, and custom manufacturing systems are a few examples of highly specialized applications that tend to be built around specific hardware and software.

In addition to limitations imposed by hardware and software, there are equally important limits imposed by regulations, certification, human factors, time frames, and cost. Regulations may require that a legacy database be accessible for another decade. Porting such an application can be cost prohibitive. A certified software system may restrict the application to a specific version of software and operating system. Application changes are not allowed to avoid a costly recertification. A replacement system may be under development, but legacy hardware has begun to fail before the new system can be deployed. Thus, an interim solution is needed to bridge the gap. Key individuals may refuse to give up a legacy system and be in a position to force compliance with their wishes. A means must be found to continue to support the application.

Of course, a key consideration is always cost. The more costly it is to replace an important legacy system, particularly one that works well, the more impetus there is going to be to find a way to maintain that system rather than replace it.

Until recently, failing hardware forced the replacement or abandonment of legacy applications, no matter how relevant and stable the software remained. Replacement was usually a costly and time-consuming proposition and sometimes resulted in the failure of a business. Now, however, a new breed of solutions is available. Know as hardware emulators, these are hardware and software solutions that replicate the functionality of specific legacy hardware while running on modern computer systems. In essence, they permit a box swap, allowing the replacement of legacy hardware with little or no change to the software or interfaces the system supports.

The hardware emulators come in two types: software based and hardware based. Both types have this in common: within the emulation environment, the target CPU, memory, disk drives, and common I/O interfaces are precisely replicated, allowing binaries from the real hardware to be ported and run on the emulated hardware *without change*.

Software-based emulators exactly replicate specific hardware using software. They run as an application under another O/S. Commercial software-based emulators usually run under

Windows while freeware efforts tend to support Windows, Linux, and others. *Table 1* provides examples of software-based emulators.

Hardware-based emulators include a hardware component that works in conjunction with specialized software to replicate legacy hardware functionality. The hardware component can take the form of a PCI add-on card for a server or a complete system housed on a single board computer. These emulation solutions tend to be highly specialized and a bit more costly, but again offer binary compatibility, functioning as an identical replacement for the original system. *Table 2* provides examples of hardware-based emulators.

Hardware-based emulators generally offer an integrated legacy bus that supports legacy control cards. Some software-based emulators also offer the means to include a legacy bus adapter. In the end, the emulator of choice comes down to what best fits the specific installation.

Thus, it is no longer necessary to discard valuable legacy applications just because their underlying hardware is wearing out. This by itself has substantial value. In addition, moving to an emulator confers further benefits.

Since emulators run on modern hardware, they help fulfill the goals of many consolidation efforts. These include consolidation to a common hardware platform, reduction in hardware maintenance costs, and reduction in infrastructure costs like space and power consumption. Furthermore, modern hardware is faster, offers greater storage capacity, improves archiving options, and may offer additional protection in the form of new redundancy options. Finally, new hardware means improved reliability.

While performance improvement is generally not a primary goal when deploying an emulation solution, it is worth highlighting because it provides often unanticipated benefits. Everything from operating efficiency to data acquisition accuracy improves with increased performance.

Probably the key benefit of an emulation solution is ease of deployment. With proper preparation, these solutions can be deployed in a matter of days and sometimes even hours. Usually, the most difficult and time-consuming task when deploying an emulation solution is transferring data from the legacy system to the emulation platform. Operational disruption is minimized because emulation systems do not change user interfaces. For all intents and purposes, they are hardware swaps. The speed and ease of an emulation solution deployment makes it a good value.

A word of caution here. Ease of deployment does not equate to easy installation. A thorough knowledge of the legacy system and emulated equivalents is required to correctly configure and install an emulation solution. This is where the "value added" in VAR means something. Specialized knowledge and experience are required to install an emulator correctly.

Hardware emulators do have weaknesses. One is that the older a system is, the harder it can be to emulate. Older hardware tends to be less standardized and contains more unique components. A good example is the HP1000. These systems were built on a near custom basis and one must examine the CPU board at the PROM and ROM level to determine if an emulated solution is viable. Conversely, Digital VAX systems were highly standardized, making them excellent candidates for an emulation solution.

Another issue with emulation solutions is software support for the legacy O/S and applications. The perception that by deploying an emulator a reduction in software support costs is realized is false. The knowledge needed to run and maintain the legacy applications must be retained. Here

the emulation solution provider plays an important role. Many emulator VAR's offer legacy application support services.

To summarize, hardware emulators make a powerful new tool available to IT departments. They permit the retention of important legacy applications, along with all of the critical business functions, investments, and knowledge these applications entail. They are quick and easy to deploy, unlike replacement, migration, or re-engineering solutions. They help achieve many consolidation goals, such as common operating platforms and reduced hardware maintenance costs. They improve application reliability and performance.

Hardware emulators do not decrease operational complexities or software maintenance costs. The legacy applications they support require retention of legacy skills and knowledge.

So the next time the hardware under an important legacy application needs to be replaced, take a look at hardware emulators. They offer a new way to use past success to build a better future.

Table 1: Examples of Software-based Emulators

Software-based Emulators		
SimH	**Data General**: Nova, Eclipse **Digital Equipment Corporation**: PDP-1, PDP-4, PDP-7, PDP-8, PDP-9, PDP-10, PDP-11, PDP-15, VAX **GRI Corporation**: GRI-909 **IBM**: 1401, 1620, 1130, 7090/7094, System 3 **Interdata (Perkin-Elmer)**: 16b & 32b systems **HP**: 2116, 2100, 21MX **Honeywell**: H316/H516 **MITS Altair 8800**: 8080 & Z80 **Royal-Mcbee**: LGP-30, LGP-21 **Scientific Data Systems**: SDS 940	The Computer History Simulation Project
TS-10	**Digital Equipment Corporation**: PDP-10 (KS10 & KL10) PDP-11 (KDF11 & KDJ11) VAX (MicroVAX II & VAXserver 3900)	SourceForge.net
ES40	**Digital Equipment Corporation:** AlphaServer ES40	SourceForge.net

Table 2: Hardware-based Emulators

Commercial Hardware-based Emulators		
Product	**System Emulated**	**Producer**
Hawk	**Data General**: Nova, Eclipse	Strobe Data
Kestrel	**HP**: HP1000	Strobe Data
Osprey	**Digital Equipment Corporation**: PDP-11	Emulators International/Logical Company