

## Overcoming 64-bit Porting Hurdles

### **S7 Software Solutions eases application porting and migration towards more stable and cost-effective platforms.**

Many companies are porting their application software to affordable high performance computing (HPC) platforms, motivated by increasing performance and creating new market opportunities. The main objective of any porting effort is to create an application on a target platform that behaves like the original version. Following transformation, the application runs on the new platform, making it available to an expanded market.

Porting legacy enterprise applications to 64-bit platforms can be non-trivial, prompting companies to seek the services of migration specialists. These specialists typically have experience converting millions of lines of code, anticipating many of the challenges involved with migrating software structures. Using specialists often saves companies time and money over doing the porting themselves.

Most 32-bit code can run in a 64-bit environment, as long as the underlying operating environment supports it. But sometimes applications outgrow their 32-bit environment limitations, requiring more addressable memory and processing power. In these cases, the code requires modifications to instructions, data structures and software constructs. However, these changes must be carefully evaluated and tested; otherwise they can render the legacy code non-functional.

Bristol Technology, a leader in business process monitoring software, wanted to port its WIND/U application, consisting of more than 2 million lines of code, to 64-bit Intel® Itanium® 2 architecture. WIND/U is a development tool that enables companies to port their Microsoft Win32/MFC based applications to leading UNIX\* and Linux\* operating systems.

S7 undertook the project of porting Wind/U on to Itanium HP-UX 11i. With rich experience in porting and migration, S7 knew this porting would not be plain recompiling and building. Getting COM working was a critical task.

#### **Challenge**

- Migrate over 2 million lines of Wind/U Code on to Itanium based HP-UX 11.
- Get COM working on the target platform. Solution • Deploy software on HP-UX platform. Run WIND/U on 64-bit Dual-Core Intel Itanium processors and the HP-UX 11i v2 operating system.
- Tweak the C structures to match the C++ class layouts.
- Use global segment address and pseudo descriptor address as padding in the C structures.

#### **Solution**

- Deploy software on HP-UX platform. Run WIND/U on 64-bit Dual-Core Intel Itanium processors and the HP-UX 11i v2 operating system.
- Tweak the C structures to match the C++ class layouts.
- Use global segment address and pseudo descriptor address as padding in the C structures.

#### **Code Porting Considerations**

Porting enterprise applications to 64-bit platforms requires an understanding of the underlying hardware and software architecture. "Porting from a 32- to 64-bit environment is not a trivial task. It requires careful consideration of your current code's use of sized constructs and the impact they can have in the new environment," says Chandrashekar, Director and CTO, S7 Software Solutions.



Although application porting can be a messy undertaking, there are some substantial benefits in doing so. Bristol Technology recognizes that having a single code base that can be used interchangeably between Linux, Unix and different hardware architectures provides opportunities to scale the solution as well as serve a broader customer base. In addition, having all their application software "under one roof" reduces development and maintenance costs for their software projects.

### **Code Porting Mechanics**

From a hardware perspective, 32-bit computers manipulate data and execute instructions as 32-bit entities, whereas 64-bit computers work on 64-bit chunks at a time. As a result, the software structures – pointers, arrays, variables – created for 32- and 64-bit operating systems and applications may be different and result in inconsistent software execution. A key challenge in porting software is to anticipate these differences, address them all at once and avoid fixing them in an ad hoc fashion.

### **Evaluating Virtual Table Layout**

Without workarounds, certain operations perform differently on the HP-UX platform than on Bristol Technology's native 32-bit platform. One such porting challenge is the inability to pass C++ objects to C routines and vice versa. Objects such as functions or subroutines are accessed through a pointer (memory address), but in some cases, there are dramatic differences in the way C and C++ handle pointers.

Wind/U contains COM code written in C language to control external interfaces such as input/output ports. This code "calls" software functions to manage COM ports (communication ports), whose addresses are accessed through pointers. However, when this COM code (C-based) executes on the HP-UX platform (C++-based), an instruction intended to fetch a pointer to the COM code instead returns a pointer to a function descriptor. Since the COM code does not receive the pointer it expects, it crashes.

To solve this problem, S7 reverse engineered the way the compiler handles the passing of pointers between C and C++ code. They identified a work around, which was to compile all the C++ code with a "flag", or compiler instruction, called "-Wc,-func\_descriptors,pseudo". This flag forces the C++ compiler to generate virtual tables that contain function pointers, so that it no longer returns pointers to function descriptors.

### **Checking For Pointer Issues**

Some of the trickiest challenges in porting native applications, such as those written in C/C++, is handling pointers. In a 32-bit environment, pointers are often hardcoded and treated as 32-bit integers. In a 64-bit environment, pointers become 64-bit values.

Wind/U was originally written with a 32-bit environment in mind. With 64-bit architecture, many software constructs, like pointers and long integers, become 64-bits wide. Long integers are program variables that can store positive or negative whole numbers. To ensure reliable code porting, S7 checked and cleaned the code wherever a pointer was concerned.

It is common for some developers to treat pointers and integers (especially long integers) as having the same size when writing 32-bit code. Before performing arithmetic operations on a 64-bit pointer, the legacy program may first assign it to a 32-bit variable and then manipulate this value. Following the execution of some arithmetic operations, the 32-bit variable is stored back into the original 64-bit pointer. Whenever a pointer is copied into a 32-bit variable, it is susceptible to truncation because half of the bits will be dropped and lost. Any piece of code accessing a truncated pointer value runs the risk of crashing the system.

### **Summary**

Wind/U is successfully ported to Intel Itanium 2 architecture running the HP-UX operating system, and it is deployed by a leading telecommunications company in Europe.

Wind/U supports a set of libraries and utility programs that enable software developers to compile, link and execute your Microsoft Win32/MFC based application on UNIX, OpenVMS, and O-/390 platforms. Wind/U accomplishes this

by supporting the Microsoft Win32 API and MFC on the X Window System and Motif, the standard graphical environment used on UNIX, OpenVMS, and OS/390.

### **Key Technologies**

- Wind/U development tools enable software developers to compile, link and execute Microsoft Win32/MFC based applications on UNIX, OpenVMS, and OS/390 platforms.
- Wind/U development tools run on Dual- Core Intel Itanium 2 based-servers and the HP-UX operating system with proven, mission-critical capabilities.

### **Integral Answers**

- S7 worked with Bristol Technology to facilitate passing C++ objects to C routines and vice versa.
- S7 identified situations where pointers would not port properly, and they checked and cleaned all application code involving pointers.

### **Motivations Behind Porting Effort**

- A key requirement of Bristol Technology's customers is to run Wind/U on Intel Itanium 2-based servers. This allows them to take advantage of a high-performance 64-bit architecture and a leading operating system.
- The HP-UX Itanium platform allows Wind/U to scale to enterprise class levels.
- The Intel Itanium 2-based solution meets customer expectations for high performance, quality and uptime.

© 2007 Itanium Solutions Alliance. \*All other names and brands may be claimed as the property of others.

*Note: Information and claims herein are provided by the award recipient and in now way warranted or endorsed by the Itanium Solutions Alliance. The Itanium Solutions Alliance does not control, verify or audit such information or claims and encourages all customers to independently obtain more information about the products.*

Copyright© 2007 Itanium® Solutions Alliance. All rights reserved. Itanium® is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.