



S7 helps port Solaris code to Itanium HP-UX 11i and Red Hat 7.x Itanium version.

Client overview

The client is a multi-regional designer, manufacturer and supplier of telecommunications and information technology equipment and services. The company's network technology and services enable its customers to evolve narrowband networks to next-generation broadband networks.

Business Need

The client had an application that was running on Solaris. Due to the constant demand of their customers, they wanted to shift the application from Solaris to Itanium.

Challenge

The software was running on Solaris 10 on a Sparc workstation and also on an Intel version (Solaris x86) platform. The software was more than 2 millions of lines of code and was written primarily in C/C++. The MFC classes including Ole/COM code with GUI were mainly coming from X/Xt/Motif. As such there was no database used but there were classes, which were used to connect, and access databases based on ODBC protocol.

The S7 Approach

S7 followed the 4-D methodology (Discover, Define, Develop, and Deploy) for this migration effort.

Discover Phase

Since we were very familiar with the software package that had to be ported, we had no issues as such with the source or target application. But we had to focus completely on the Itanium machine and the O/S on it i.e. HP-UX 11i and Red Hat Linux.

First we decided to migrate the application onto HP-UX 11i and then on to Redhat Linux. We wrote a few similar applications on HP-UX 11i and then compiled both on to Solaris and Itanium and started out from there. By then we had compiled the complete migration issues from Solaris to Itanium HP-UX11i and also from Solaris to the Itanium version of the Redhat Linux.



S7 Software Solutions
Re-defining Cross-Platform Porting,
Migration & Development

Define Phase

We ran our parser to analyze the source code and list out all possible porting and migration issues. We also made a list of all possible system and third party dependencies and figured out the location and correct versions on the Itanium machine for system libraries. We also got all required third party libraries for Itanium versions.

There was an issue with selection of libraries which one uses for linking, as there is both a 32-bit and 64-bit version of these libraries. We linked with the 32-bit libraries and selected all the right ones. We wrote small applications to use them, link them, etc to get a clear picture of their usage and linking; and to confirm that they were the right replacement libraries.

Basically we were completely ready as far as configuration was concerned for the Itanium migration. We also created the appropriate arch files with the right flags that aided us in the compilation and debugging when required.

Transformation Phase

We copied the source code completely onto HP-UX11i and then started compiling the code. We mainly hit a lot of ANSI non-compliant issues while compiling using aCC because aCC is strict ANSI compliant. Once all those issues were fixed, we hit a few undefined kernel and system calls because there were slight differences with the naming or the number of parameters or with the path of the include files. We also hit a few issues with the flags that we had to use for compiling but since we had the requisite knowledge and experience, it became easier.

Although we got the linking all done successfully, we hit many runtime issues and most of them were mainly related to failed name resolution for which we had to find the equivalents for Itanium. The biggest roadblock we hit was the [virtual table accessing](#) that changed considerably in Itanium.

This was a tough job and we worked with the HP Compiler group and also debugged in the assembly level as the vtable layout had changed completely. Yet again, we got through it and completed a successful migration.

Once this was complete, we copied this code on to the Redhat Linux Itanium version and started compiling. Other than a few name changes, structure changes, include directory changes and vtable tune up, the whole migration was a breeze.

A good thing that we did was to have parallel development of the complete test suit including the acceptance criteria. We ran all test cases and got it to a complete stable form ready for deployment.

Deploy/Delivery Phase

The deployment was a bit tricky as we had to go on-site (UK) for the deployment and fortunately since a lot of testing work went in we had no surprises and everything went off smoothly as planned. The project was signed off and thus we finished a complete successful migration of the software to Itanium for both HP-UX11i and Redhat Linux.