Apple Backs Binary Translation Tool for PowerPC

Echo Logic "FlashPort" Converts 68000 to PowerPC Binaries

By Brian Case

Apple and Echo Logic announced on May 4 a collaborative effort to help Macintosh application developers port their programs to the PowerPC architecture. Apple wants a large suite of native PowerPC applications on announcement day of the first PowerPC machine—rumored to be 12 to 18 months away—and Echo Logic has the technology that will make it easier for application developers to oblige.

FlashPort, which is Echo's first product, is a sophisticated, automatic static translation tool. It takes as input the binary 68000 version of a Macintosh application and outputs a native, PowerPC version. With FlashPort, developers should be able to complete a PowerPC native port of an application in as little as a few days or at most a few weeks. Since few human resources are required to accomplish the port, a developer can focus most of its resources on adding features to new PowerPC-only versions of its programs that take advantage of the performance of PowerPC.

While the results will vary depending on the application, Echo expects translated applications to achieve 90% of the performance of native compiled applications, and to have only a 10% size penalty. These goals have yet to be demonstrated, however.

While FlashPort might require little in the way of human resources, it is a hardware hog: translation must be done on an RS/6000 workstation with between 64 and 128 Mbytes of RAM (costing about $20,000). Since developers may balk at the expense of such a machine just for a two-week porting project, Apple is considering setting up an RS/6000 leasing program.

Though officially formed only recently, Echo Logic's engineers have been developing the technology for five years, which reflects the ambitious nature of their undertaking. Echo is the result of a research project that started with three compiler engineers at AT&T Bell Labs, and Echo is an AT&T-funded venture. The Echo founders include chief scientist Christopher Macey; chief technology officer John Goettelmann; president Bradford Burnham; and COO Douglas Sabella.

The researchers purposely chose the Macintosh as the test case for their binary translation techniques because of its difficulty: the system and toolbox routines are mostly hand-written assembly code, as are some significant applications. After a couple of years, they succeeded in proving the techniques workable.

Today, the choice of Macintosh as the test case has proven especially fortuitous. After a demonstration of the results to Apple engineers, Apple committed to use the technology as a key component in the transition from 68000- to PowerPC-based computers.

FlashPort Overview

Echo Logic is in the business of selling and supporting binary application porting tools that will be used by the original developers of applications. With FlashPort, the original developers will be able to quickly produce versions of applications that run on a new host computer. Furthermore, the ported application will run on the new host as a native application, using the native ABI (application binary interface, which refers to object code file format, conventions for calling system services, stack frame construction, etc.).

FlashPort is delivered from Echo in two forms: a complete application translator and a fragment translator. Though it is probably easiest to simply use the complete translator, the fragment translator can be used when an application consists of both HLL and assembly code; the HLL can be recompiled in the standard way and the fragment translator used to do the nasty job of converting source assembly code to native assembly code for the target processor.

How FlashPort Works

FlashPort works by looking at the binary code for an application as a form of source code. The first job, performed by the analyzer phase, is to find all the code in the executable file. This is done by starting at the program's pre-defined entry point and following the flow of control both sequentially and through relative jumps and subroutine calls.

When the Analyzer finds indirect jumps and calls—those whose targets depend on data values at run time—symbolic execution is performed. (Symbolic execution is like interpretation except that symbolic values are used for register and memory contents instead of real data.) Although this is a powerful technique that can successfully handle nearly all indirect-jump situations—including even elaborate tables of branch addresses—it can fail under certain circumstances. For this reason and others, FlashPort allows interaction with the porting engineer to gain high-level knowledge about the program to improve the analysis.

The process of finding all the code produces a very rich, detailed program description that includes a complete flow graph and verbose semantics for each source machine instruction. At this point, the representation is so detailed that it can require 100 times the space of the original program.

The resulting program description is the perfect target for optimization algorithms; it is similar to a RISC machine-language program. Standard textbook and specialized optimization algorithms are used to eliminate most of the semantic description for each instruction. For example, the description for a 68000 arithmetic instruction includes explicit information about how to set each condition code bit. In the vast majority of cases, the semantics for all the condition code bits can be eliminated because they are never used.

Just as in a standard optimizing compiler, the optimization phase is followed by a code generation phase. In FlashPort, the code generator is responsible for adhering to the architectural conventions, such as the procedure call mechanism, of the target machine. It also takes care of things like delayed branches.

The code generator produces both a human-readable assembly language file and a linker-ready object file. The native object-code linker is then used to produce an executable native application.

One very helpful aspect of FlashPort is its ability to carry debugging information from the source application to the translated native application. Where possible, FlashPort creates a native application that contains native-format symbolic debugging information that allows native debugging tools to be used.

For example, Macintosh programs might be debugged with the popular TMON debugger; for UNIX-based PowerPC systems, FlashPort will be able to generate debugging information compatible with the popular DBX debugger. Some of the aggressive optimization algorithms will make debugging more difficult (as with all optimizing compilers), but an experienced engineer will be able to overcome the limitations.

Comparison With XDOS

Hunter Systems has had its XDOS PC-to-UNIX translation product on the market for some years already, but Echo is not going to compete directly because of a different marketing approach. Hunter Systems' porting tools are used only by Hunter Systems engineers. Application developers either pay Hunter Systems for the porting services or license their application to Hunter Systems. In the case of licensing, Hunter Systems creates a native version of the application and serves as distributor for the UNIX version.

In contrast, Echo is only in the business of developing and selling the porting tools. Their plan is to sell tools to the original software developers. These developers then sell the re-hosted versions.

While the underlying techniques of FlashPort and XDOS are probably very similar, contrasts in company history, marketing strategy, and underlying computer environments result in differences in the tools.

FlashPort's engineers had the relative luxury of a research environment. They had years of time to contemplate problems and try various solutions. XDOS' creators, on the other hand, had the schedule and financial pressures of a startup environment to consider. They needed to balance the technical problems with marketing considerations and time constraints.

Perhaps even more critical is the nature of the source and target environments. Echo chose the Macintosh, which, while written in hand-crafted assembly language, provides a very structured, relatively well-behaved framework. The Macintosh environment is event driven, so applications must, for example, make a "wait-next-event" call when they are idle.

In contrast to the Macintosh, the PC environment presents a variety of additional problems. For example, a busy-wait loop is used by PC applications when they are idle. Since the UNIX target chosen for XDOS is multitasking, a busy-wait loop would be inexcusable.

XDOS also must simulate PC system services using native UNIX resources. Apple will provide FlashPort with native versions of Mac ToolBox and OS calls. The upshot is that FlashPort appears to be a much more automatic tool.

Over time, Hunter Systems has modified its marketing approach to accommodate the desires of its clients. Early on, they were pushing a plan that relied on selling "key files" (files of the human-generated hints needed to complete a translation). Hunter Systems is probably going to push into the Windows market in a big way. Since the Windows environment is well-structured, like Macintosh, they may be able to make XDOS as automatic as FlashPort, and key files may again become an important part of their strategy.

Market Potential

In the wake of all the activity surrounding the PC market, it is easy to forget that the billion-dollar markets of yesterday were founded on minicomputers and mainframes. Some of the applications developed for these machines are still important.

Thus, Echo has a potentially diverse market at its feet. Even if the PowerPC-based Macintosh market falls below expectations, Echo can probably thrive by supplying its technology to those hapless companies drowning in millions of lines of assembly code. There is also the PC market, as Windows NT becomes available on MIPS and Alpha. Microsoft's public strategy is to use emulation until full native versions of Windows applications are available. If Windows developers are impressed enough with the results of the 68000-to-PowerPC transition, Echo may have a big opportunity to help developers generate MIPS and Alpha versions.♦