Table of Contents Author Index Home

Get Acrobat Reader Get Postscript Version Get PDF Version

UNICOS to IRIX Applications Directions

Peter A. Rigsbee
Cray Research, Inc.
655A Lone Oak Drive
Eagan, MN 55121

par@cray.com

ABSTRACT :

The hardware and software roadmaps presented by Cray Research and Silicon Graphics show IRIX systems as the follow-on systems to traditional Cray architectures. These roadmaps, though, are at too high a level for application programmers to understand and assess the impact of these changes on their applications. This paper presents plans and on-going activities within Cray to ensure that this application migration proceeds smoothly.

KEYWORDS:
Applications, Libraries, Compilers

Introduction

Cray application programmers have made a significant investment in software targeted for Cray systems, and additional investment continues today. The hardware and software roadmaps presented by Cray Research and Silicon Graphics show IRIX systems as the follow-on systems to traditional Cray architectures, starting with the "SN1" system identified as the follow-on to the CRAY T3E. Customers seeing these roadmaps have raised critical concerns about the impact of these changes on their application investment.

Cray Research and Silicon Graphics recognize the importance of ensuring that current Cray applications port easily and effectively to these new architectures. This issue has received significant attention, at all levels of the company, from the very beginning of the merger discussions.

In the year since the merger, these discussions have evolved into clear statements of goals and requirements, and, more importantly, into numerous development activities planned to implement these goals well before the release of the first SN1 systems. The remainder of this paper discusses these goals, requirements, and development activities, both for UNICOS to IRIX migration in general, and for T3E to SN1 migration as the initial requirement.

Goals and Requirements

Application programmers porting a code to a new platform have two key goals they want to achieve:

- complete source code compatibility
- full exploitation of performance characteristics of new system

Cray Research and Silicon Graphics share these goals, and have hardware and software projects underway to deliver products that meet these. Restated from a Cray perspective, these goals become:

- programmers should be able to write portable applications for UNICOS, UNICOS/mk, and IRIX systems with no source code differences
- when a customer moves from a legacy Cray system to its follow-on IRIX system, this should be accompanied by better performance

When looking at a more detailed level at migrating applications from the CRAY T3E to SN1, the above goals apply, and a couple additional requirements were identified. First, in order to deliver better overall performance, it is necessary that the performance profile of the SN1 system be similar to that of the CRAY T3E. Parallel applications performance results from a balanced combination of single-PE performance and parallel behavior and performance. If two systems have a significantly different balance here, then an application is likely to require significant retuning or restructuring to get good performance on the new system.

Second, application development tools designed for large parallel applications are critical. The CRAY T3E has such a set, and it is necessary that comparable (if not better) program development tools be delivered with SN1.

Now to understand migration issues, it is necessary to look at both the hardware and software plans.

T3E and SN1 Architectural Comparison

The CRAY T3E and SN1 are described using different terms and names, and the two systems evolved from different directions. The CRAY T3E was designed as a scalable, distributed memory system evolving from the similar CRAY T3D, whereas the S2MP architecture used in the SN1 and introduced with the Silicon Graphics Origin computer systems, evolved as the scalable follow-on to Silicon Graphics' earlier shared-memory servers. How then, some people ask, can SN1 serve as a follow-on to the T3E? Cray Reserach and Silicon Graphics believe it is very well positioned as a follow-on, and will easily support and deliver good performance for applications developed initially for the CRAY T3E or T3D. And it goes even further, offering options and flexibility not available in the Cray systems.

When Cray Research initiated the Cray MPP project in the early 1990s, the company recognized that the computer industry in general, and scalable computing in particular, were in a state of rapid and frequent evolution and change. In order to provide Cray's engineers with the ability to respond to and exploit changes, Cray introduced the "Cray MPP macro-architecture". This characterized the key aspects of the Cray MPP systems that would be carried forward from the CRAY T3D to the T3E to its follow-on system. Other attributes of the system -- the micro-architecture -- could change, but the macro-architecture would be maintained. A program designed to exploit the macro-architecture would be need little tuning as new Cray MPP systems were introduced. This approach was used very sucessfully between the CRAY T3D and CRAY T3E, where significant low level changes were made (most notably the remote memory access method) without disturbing the macro-architecture or user interface.

What was this macro-architecture, and are these attributes met by the S2MP architecture and SN1 system?

- **MIMD execution with SIMD synchronization**. The CPUs in SN1 run independently, but the system includes a hardware barrier providing SIMD synchronization.
- **Globally-addressable distributed memory.** The distributed shared memory of the S2MP architecture takes the globally-addressable distributed memory of the CRAY T3D and T3E a step further, providing a single address space across the entire machine.
- **Fast data interconnect.** SN1 will have a low-latency, high-bandwidth data interconnect.
- **Fast synchronization and control.** With a hardware barrier and synchronization constructs associated with its shared memory, SN1 meets this requirement.
- **Support single application running on up to thousands of processors.** The S2MP architecture was designed for scalability, and the SN1 system will scale to thousands of processors.

Now consider the work that a programmer needs to do to get an application to scale well on a CRAY T3E or other MPP system. To achieve this, the first step was to make sure the application had a very high degree of parallelism. Then, data decomposition was performed to minimize the amount of data communicated between tasks. And a likely third optimization step was to reduce the number of messages used to communicate this data.

The key point to recognize is that a program written in this manner is ideally suited to exploit a distributed, shared memory parallel system like SN1. Why? By maximizing the parallelism in the application, the application may now be able to use the hundreds or thousands of processors available on the system. Since memory on SN1 is physically distributed, the data decomposition step carried out for the T3E leads to efficient use of the SN1's memory. And the communications interfaces - be they MPI, PVM, SHMEM, or HPF - will be available on SN1, and internally use the fastest mechanisms in the hardware for data communications. So the distributed memory part of the application will port well from T3E to SN1.

And SN1 has some capabilities not present on T3E. For example, most applications have some serial component, and on SN1 this component can directly access memory used by the entire application, without having to be aware of whether the memory is local or remote. As another example, there are often parts of an application that will not scale well, or the effort required to manually parallelize the code is excessive compared to its benefit. With SN1, the application programmer can choose to use automatic compiler parallelism on these portions of the application to get some performance improvement, with very little effort.

Other capabilities of SN1 not found on CRAY T3E stem from the Silicon Graphics heritage of the system. The means that SN1 customers will have thousands of MIPS-compatible applications available, and have access to a much wider range of graphics libraries, languages (such as Ada95 and Java), and other development tools than were available on the CRAY T3E.

So from an architectural perspective, SN1 provides the basic performance attributes and capabilities expected by the current CRAY T3E user, plus more.

Migration Software Strategy

The Cray Software Division strategy for supporting application migration has two key components.

First, application source code compatibility between IRIX and UNICOS is being given very high priority. A project called the "Supercomputing API" has been initiated to identify the language and library features that will be supported on both UNICOS and IRIX systems, thereby defining the language for HPC application development. The next section of this paper will talk in more detail about the Supercomputing API.

Second, the Cray Origin2000 system is being used as an early-delivery platform for UNICOS and Cray features. In order to permit SN1 to serve as a T3E follow-on, it is necessary that everything be completed by the time SN1 is delivered. To achieve this, new features such as the Supercomputing API, UNICOS-like production capabilities, and features necessary to support MPP-like applications are being developed on the Cray Origin2000 system and released in near-term SGI software products. Note that this isn't the same as saying that the Cray Origin2000 can today serve as a T3E follow-on or substitute.

This early-delivery approach gives the company and customers the opportunity to get real experience with these features and capabilities well before Cray customers will be moving in large numbers to the new architecture. Cray has a 128-CPU Origin2000 installed in Eagan, for this very use. Work is underway today, using a mixture of released and prototype software, to understand first-hand the issues around application migration, performance (both application and system-wide), production operations, and the development and release process.

Supercomputing API

Applications Programmer Interface (API) is a term used to refer to the set of the library functions or language features for use by applications programmers. Over the years, Cray and Silicon Graphics have created APIs for their UNICOS and IRIX systems, respectively; these APIs have evolved as new languages and libraries were added, or as enhancements were made to existing languages and libraries. The Supercomputing API is an effort by Cray and Silicon Graphics to select the set of features that will be available on UNICOS **and** IRIX computer systems, and that can be used as the basis for high performance application development. An application programmer who uses features from the Supercomputing API will know that he is using features that will be available across these systems, and so will thereby have one less thing to worry about when moving between systems.

The Supercomputing API does not restrict either the UNICOS or IRIX APIs; these will continue to evolve and develop to meet customer needs. The Supercomputing API is being defined now, and is planned to be made available for customer comment and guidance by summer 1997. Much of the Supercomputing API is based on industry standards that have been available on UNICOS and IRIX systems for many years; the remainder is planned for implementation over the next 12-18 months.

The Supercomputing API is a combination of industry standards, Cray extensions, and Silicon Graphics extensions. Industry standards, including formal standards such as Fortran 90, C++, C, and X/Open XPG4, and de facto standards such as MPI, PVM, BLAS, and LAPACK, comprise the bulk of the API. These standards are available on Cray and Silicon Graphics systems today.

The next largest set of features are based on Cray extensions to languages and libraries. These include the key Fortran and C extensions, and many key library extensions, including the FFIO I/O optimization layer and the SHMEM one-sided message-passing library. Particular emphasis has been placed on including as many Cray extensions as possible in order to achieve the goal of source compatibility with current Cray systems. A few of these extensions are available on IRIX systems today, with many others under development.

The Supercomputing API will also include current and new SGI extensions. Currently this is the smallest of the three groups, consisting of some math functions and some multiprocessing language directives, but is expected to grow as more Cray customers begin to use IRIX systems and identify features of interest that would be important to have implemented on UNICOS.

There will be some Cray features not in the Supercomputing API. These will typically fall into two categories - obsolete and architecture-specific. Obsolete features are things like library packages (such as LINPACK) that have been replaced by more modern packages (such as LAPACK). Certain Cray-specific vector functions are also likely to be omitted. To help application migration, detailed information will be provided on alternatives for code that use these features. These features will continue to be supported on UNICOS systems.

Software Product Plans

Plans are important, but must be followed by actions. As soon as the merger between Cray and Silicon Graphics was announced, planning was followed quickly by development activity in several key areas, and this has increased over time. For application migration, the key areas of concern are:

- compilers (for Fortran (77 and 90), C, C++)
- libraries (especially scientific, I/O, and message-passing)
- tools (debugger, performance and static analysis tools)

The remainder of this section will cover major activities in each of these areas.

Compilers

The most important activity in the compiler are related to application migration is the use of Cray's CF90 front-end in the upcoming MIPSpro Fortran 90 7.2 release. The CF90 front-end, supporting Cray language extensions, is being combined with the MIPSpro back-end, designed for MIPS code generation. This single effort will address many of the source compatibility issues important to Fortran applications.

Cray introduced several extensions to its Standard C and C++ compilers, most of which will be implemented in MIPSpro compilers. And a very important, albeit less visible, compiler activity, is the functional and performance testing that Cray developers have been carrying out on MIPSpro compilers and on Origin systems.

Libraries

Three key library efforts are worth noting. In March 1997, Silicon Graphics saw the release of SCSL 1.0 -- the Silicon Graphics/Cray Scientific Library. This joint development project, which began soon after the merger, is a new scientific library initially released for IRIX 6.4 systems (including the Origin line), and which is expected to be released for Cray systems with IEEE arithmetic. SCSL 1.0 includes versions of BLAS, LAPACK, and FFTs, optimized for Origin systems.

Cray's innovative FFIO (Flexible File I/O) package is a set of library functions that use the 'assign' command to deliver I/O optimizations and data conversion in a very convenient and extensible manner. This package has been ported to IRIX, and the first release, which includes many of the Cray FFIO "layers", will be part of the MIPSpro 7.2 compiler releases.

Message-passing is extremely important to CRAY MPP customers, but has never received high visibility within Silicon Graphics. This is unfortunate, because Silicon Graphics has been a major contributor to the MPI Forum, and has a high quality MPI implementation. A new product, called the Message-Passing Toolkit (and modeled after the Cray product of the same name) has been defined for IRIX systems, and will soon have its first release. MPT will include optimized implementations of MPI, PVM, and SHMEM for IRIX systems. Later releases will see the Cray and SGI MPT products move towards the goal of becoming a single, interoperable product.

Tools

To meet future requirements, especially for high-end customers doing development of large, parallel applications, a new development environment is being developed for IRIX systems. This joint project, code-named "Caribou", involving people from Cray and MIPS (SGI), will produce a truly integrated development environment, handling debugging, performance analysis, and static analysis. The new product will draw from both CrayTools (Totalview, MPP Apprentice, ATexpert, etc.) and SGI's WorkShop product. Special emphasis has been placed on meeting high-end requirements in the first release of the product.

Summary

Cray Research and Silicon Graphics are committed to the easy migration of applications from current Cray systems to the follow-on systems in the product roadmap. Initial focus is on the CRAY T3E to SN1 transition, but many of the same issues and most of the development applies to transitions from any Cray system. The company believes that SN1 will be well-suited as an effective follow-on for the CRAY T3E. The Supercomputing API will allow technical HPC applications to be developed in a manner that allows their use on both UNICOS and IRIX systems. And finally, significant development is underway within the company to deliver on the plans.

Author Biography

Peter Rigsbee is the Software Product Manager for Cray Programming Environments and Developer Software in the Cray Marketing Division, a position he has held since December 1996. An employee of Cray Research since 1980, Peter has held a variety of technical and management positions, primarily in the Software Division, including such areas the CRAY T3E software project, PVM and MPI message-passing software, and Cray debugging and performance tools.

par@cray.com