# The CraySoft CF90 Programming Environment: A User's Perspective

*W. Robert Boland*, Distributed Computing, Los Alamos National Laboratory, Los Alamos, NM

**ABSTRACT:** *In May 1994, CraySoft released their CF90 Programming Environment for SPARC systems to selected beta test sites, one of which was the Los Alamos National Laboratory. In September 1994, version 1.0 was released. One of CraySoft's stated objectives is to provide programmers with a common Fortran development platform on the PVP and workstations. We shall describe our experience with this and similar products. We have compiled a number of codes on a Sun SPARC with the CraySoft f90 and the Sun f77 (where appropriate) compilers and then run the executables. The exercise has been replicated on a Cray Y-MP with the Cray cft77 and f90 compilers and an IBM RS6000 model 590 with the XLF Versions 3.1 and 3.2 compilers. We shall report on such metrics as compile time, quality of error messages, size of the executables, and execution times. As time permits, we shall discuss other aspects of the CF90 Programming Environment such as ATExpert, TotalView, cflist, cflint and xbrowse.*

## Introduction

Shortly after the CraySoft CF90 environment was publicly released, LANL ordered a small number of licenses for production as well as continued evaluation use. Several users had expressed the desire to develop codes on a SPARC platform and later move the code to either a Cray PVP or MPP for final check-out and production. The applications run the gamut from relatively small to tens of thousands of lines of code.

We also have a large collection of codes written primarily in Fortran 77 which have been used in the past for evaluating both Fortran 77 and Fortran 90 compilers. It should be noted here that when we use the term Fortran 90, we mean a code which adheres to the Fortran 90 standard and includes at least some Fortran 90 language constructs which are not in Fortran 77. The evaluation of the CraySoft compiler was one in a long series of evaluations which we have done over the past number of years and the second test of a Fortran 90 compiler.

Our approach in this paper is to report primarily on our experiences with the CF90 environment. Although performance (compile time and execution time) are certainly very important, that will not be our emphasis here. Also, the CF90 product is a very immature product at this time and we feel that execution timings are not as important as features and the ability, for example, of the compiler to compile standard conforming code correctly.

A recent event which significantly changed this paper and our talk was the release of version 1.0.1. For some unknown reason, we did not learn of the existence of this release until early March 1995 and only installed it within the past few days. Because of the improvements in this version, we felt it inappropriate to present results for an out-of-date product; this is a fast moving project and one must run as hard as one can in order to stay in place.

## The compiler f90

Certainly the primary emphasis of this paper is to report on our experiences with the compiler itself. As stated above, we will not be giving benchmark results. With that caveat, we do report that, in general, we have found the compile times to be higher using the CraySoft f90 compiler on Fortran 77 codes than those using the Sun version 3.0.1 f77. We have also compiled identical codes on the Cray PVP and the IBM RS600 workstations. The IBM workstations are model 590s. Because of the load on the PVPs, the wall-clock times from the SPARC are better than those from the PVP. The nodes of the IBM cluster have been heavily loaded recently, and a similar experience has been encountered when comparing the SPARC and the IBM workstations. It should be noted that we are using a SPARC which has a lot of idle time; often we are the only active user. A similar scenario exists in execution times.

Since the SPARC f90 compiler is still a very immature product, we do not wish to report, in detail, many of the compiler problems which we have found. We continue to find these, and fairly often. We still have a number of codes written in ANSI Fortran 77 which compile fine and give wrong answers. Due to more frequent upgrades with the CF90 product on the Cray PVPs, we are finding fewer problems there. This leads us to one of our biggest concerns with this product. CraySoft plans quarterly upgrades. Based upon our experience with

many vendor products, this is just insufficient - we must have more frequent upgrades.

In our opinion, there is one very serious problem with the compiler. This is the lack of a compiler switch for what we commonly call “automatic double precision.” Since one of the primary objectives of the CraySoft SPARC compiler is to provide the same development platform on the SPARC platforms as the Cray PVPs, one must be able to compute with (roughly) the same arithmetic on both platforms. Single precision on the Cray PVP is 64 bits and on the SPARC is IEEE 32 bit arithmetic. The IEEE standard provides for 64 bit floating point. Thus, in order to obtain comparable results, one must use 64 bit floating point on the SPARC. But, many developers do not wish to use the standard Fortran DOUBLE PRECISION declarations, etc. in their applications. Thus, for many, the only viable approach is the use of a compiler switch. It is our understanding that this will be a new feature in version 2.0.

We have many users who have found it advantageous to reference some of the system library routines. This includes getenv to obtain the value of a particular environment variable, iargc to obtain the number of arguments on a command line, getarg to obtain the nth argument, and system to run a Unix command. With the present implementation of the compiler most of these do not work as one might expect or hope. This problem is being addressed. For example, the Posix to Fortran (PXF) subroutine interface to GETARG and IARGC is standardized and is portable between systems such as the SPARC and Cray PVP systems. CraySoft will be providing a PXF interface in a future SPARC update to allow use of PXFGETARG and IPXFARGC on SPARC systems.

One feature of CF90 which we have found to be very helpful is the explain utility. Those fairly new to the Fortran language as well as very experienced developers appreciate this. Some of the messages produced (at least in the beta release) were a bit too cryptic and not well formatted; we consider this a minor annoyance.

Several developers who have tried both the CraySoft f90 and another vendors compiler prefer the CraySoft product since it allows more freedom in the Fortran 90 standard.

## TotalView

The TotalView debugger has been a source of great frustration, especially during the beta test last summer. Sometimes, we were forced to abandon it and use devices from the early days of computing, such as “print” statements, in order to find problems. With this said, we recently received a briefing from Cray on their 1995 TotalView work plan and were extremely pleased. We are optimistic about this product and look forward to receiving both the 1.2 and 2.0 releases.

## cflint

Our experience with cflint has been fairly limited, probably since many of the codes in our repertoire are already quite bug-free. On the other hand, we have used cflint on a number of occasions to find problems in user codes in our work with the local Consulting Office. It has been a valuable tool. We would like to be able to turn off some of the checks which cflint does. In some of our legacy codes, there are constructs which are not standard but acceptable by all compilers; we are well-aware of these and are willing to assume the risk. For these, it would be nice if we could turn off the checking. Also, some users have expressed frustration at the large amount of output - pages and pages of code listing with no problems noted. Give us just the facts.

Another situation arises in large code development where many of the routines are compiled and placed into a library. There we would like to be able to supply, for example, a main program and a few other modules along with the library and have cflint do the intraprocedural analysis. Many of the programming errors which we find are due to faulty referencing of modules.

As we become more familiar with cflint, we may find that the above objections have already been taken into account by the developers; in such case, our apologies in advance.

## Documentation

The CraySoft CF90 comes with the standard set of man pages and a small printed manual. We have found these to be sufficient for most purposes. For those who want more information, the CrayDoc system is available. We have installed and tested the Sun OS 4.1.x, the Solaris 2.x and Hewlett-Packard versions. Basically these work as advertised. Although some users have expressed the desire to have the documentation available via Mosaic, Netscape or Gopher, we have found a number of advantages to the CrayDoc product. For example, searching for topics below the current place holder, in our experience, has been very easy with CrayDoc.

That said, we do have a major complaint about the distribution of the documentation files. If, for example, one uses CrayDoc to provide information for the SPARC, PVP and MPP, the natural installation process results in three (3) copies of the Fortran 90 manual. If one uses the same server to deliver documentation to SunOS 4.1.x, Solaris 2.x and Hewlett-Packard workstations, there will be a total of nine copies of the Fortran 90 manual. This borders on waste, fraud and abuse. We contend that the normal installation process should use the obvious devices to point to a single copy of each document.

## Conclusion

Although the CraySoft CF90 Programming Environment is still a very immature product, it has already proven to be a valuable development tool at Los Alamos. We have found that developing applications on a SPARC workstation can be done very effectively. Moving the code to a PVP is often then a trivial exercise. We do not have sufficient experience with moving codes to a Cray T3D at this time, but we know that to be a much more difficult task.