Cray TotalView™ Development Plans

Dennis Moen, *Pete Johnsen*, *Dennis Parker*, and *Allen Stipek*,
Cray Research Inc.

ABSTRACT: *Cray TotalView is a window-oriented multiprocessor symbolic debugger. First released in late 1993, Cray TotalView is a key component in the Cray Programming Environment. This paper is an overview of 1995 Cray TotalView development plans. These plans address user interface improvements, data visualization, line-mode and scalable debugging for Cray T3D programs.*

Introduction

Cray TotalView is a window oriented debugger which is capable of debugging multiple-process, multiple-processor programs as well as single-process programs. The TotalView debugger runs under the X Window System and its graphic interface is easy to learn and good for conveying the basic concepts of parallel programs. It provides a large repertoire of features for debugging programs written in Fortran 90, FORTRAN 77, C, C++ or assembler. TotalView was designed to be portable and provides a good base for developing a debugger that spans multiple languages and hardware architectures.

Cray TotalView was initially released in the fourth quarter of 1993 to provide support for Cray T3D programs and Cray Fortran 90 programs running on Cray Y-MP and C90 platforms. In 1994, Cray TotalView 1.0 was released to support Fortran 90, FORTRAN 77, C, C++ and assembler programs running on Cray massively parallel, parallel vector and SPARC computer systems. It provides a common debugger across all Cray supported platforms, allowing programmers to leverage their learning investment when they program for more than one type of system.

This is an overview of 1995 Cray TotalView development plans. These plans address user interface improvements, data visualization, line-mode and scalable debugging for Cray T3D programs.

User Interface

The TotalView 2.0 interface is based on OSF/Motif. Motif style menus have been added across the top of all windows. Pull-down menus, buttons and dialog boxes make the functionality of TotalView more easily available to the user. The appearance of windows, menus, dialog boxes is standardized as much as possible to increase user familiarity with the interface. A command window has been added to output informational messages from TotalView and input commands to TotalView. Figure 2 shows the TotalView 2.0 process window with the top-level pull-down menus, the source window, and the command window.

Data Visualization

Data visualization gives the user an efficient means to navigate and interpret large amounts of data. Visualization is an integral part of the debugging process and TotalView's data visualizer will be simple and efficient to use while debugging. TotalView 2.0.1 will support an integrated capability for the visualization of parallel and serial data. This initial support will include a 2-D contour representation with color map or gray scale intensity mode. The visualizer is designed to display a two-dimensional slice of data and allows the user to pan around in this slice.

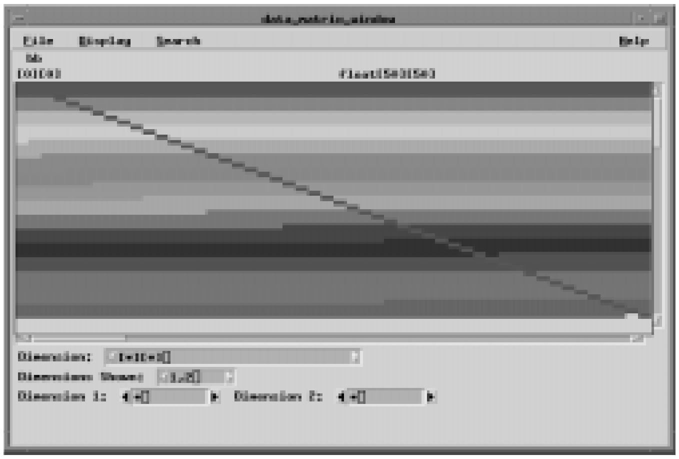
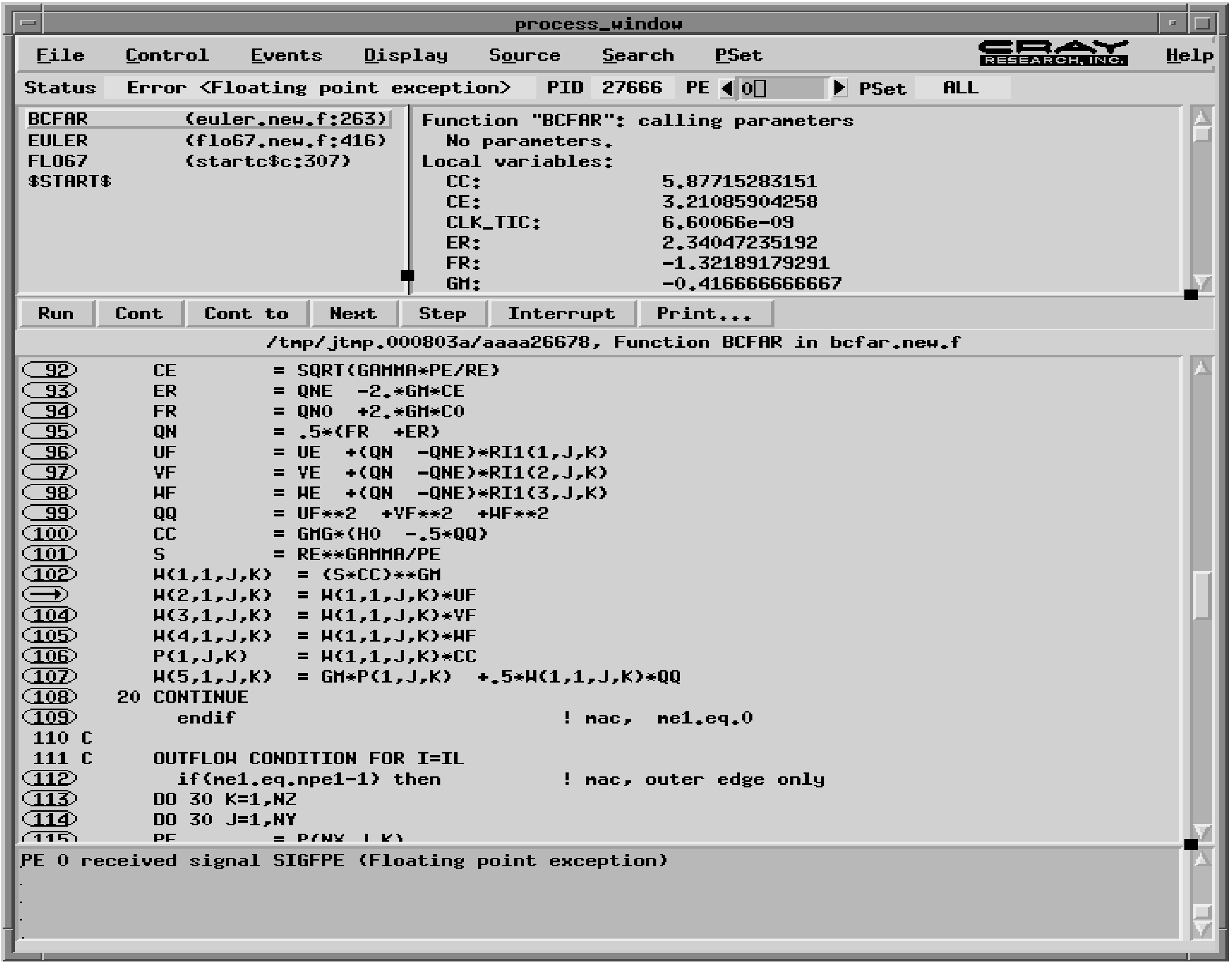


Figure 1: Array visualizer



```
Status   Error <Floating point exception>   PID 27666   PE 0   PSet   ALL

BCFAR      (euler.new.f:263)
EULER      (flo67.new.f:416)
FLO67      (startc$c:307)
$START$

Function "BCFAR": calling parameters
  No parameters.
Local variables:
  CC:                 5.87715283151
  CE:                 3.21085904258
  CLK_TIC:            6.60066e-09
  ER:                 2.34047235192
  FR:                 -1.32189179291
  GM:                 -0.416666666667
```

```
/tmp/jtmp.000803a/aaaa26678, Function BCFAR in bcfar.new.f

      CE        = SQRT(GAMMA*PE/RE)
      ER        = QNE  -2.*GM*CE
      FR        = QNO  +2.*GM*CO
      QN        = .5*(FR  +ER)
      UF        = UE  +(QN  -QNE)*RI1(1,J,K)
      VF        = VE  +(QN  -QNE)*RI1(2,J,K)
      WF        = WE  +(QN  -QNE)*RI1(3,J,K)
      QQ        = UF**2  +VF**2  +WF**2
      CC        = GMG*(HO  -.5*QQ)
      S         = RE**GAMMA/PE
      W(1,1,J,K)  = (S*CC)**GM
      W(2,1,J,K)  = W(1,1,J,K)*UF
      W(3,1,J,K)  = W(1,1,J,K)*VF
      W(4,1,J,K)  = W(1,1,J,K)*WF
      P(1,J,K)    = W(1,1,J,K)*CC
      W(5,1,J,K)  = GM*P(1,J,K)  +.5*W(1,1,J,K)*QQ
   20 CONTINUE
        endif                         ! mac,  me1.eq.0
C
C     OUTFLOW CONDITION FOR I=IL
        if(me1.eq.npe1-1) then        ! mac, outer edge only
      DO 30 K=1,NZ
      DO 30 J=1,NY
      PE        = P(NY,J,K)

PE 0 received signal SIGFPE (Floating point exception)
.
.
.
.
```

Figure 2: TotalView 2.0 process window

Line-mode

TotalView 2.0 introduces a line-mode interface. Line-mode is a commands-only mode running without the graphical interface. If the DISPLAY environment variable is set prior to running TotalView, TotalView will bring up the X Window System interface instead of the line-oriented interface. If the user specifies the -L option on the totalview command line, TotalView assumes that the user wants the line-mode interface.

Some users prefer a command-line access over a graphical one even from the X Window System interface, so in TotalView 2.0.1 a subset of commands will be supported in the X Window interface. A number of actions can be rapidly performed by typing in the command window.

For a list of TotalView commands, see the "Cray TotalView Debugger Reference Card", publication SQ-2519. To obtain additional information regarding the format and examples of each of the commands, type "help" at the totalview prompt when running TotalView in line-mode.

Table 1. Common TotalView line-mode commands

Command	Description
stop in	Stops execution when subroutine is called.
stop at	Stops execution at source line.
delete	Deletes breakpoint.
run	Executes program from the beginning.
cont	Executes program from current execution address.
step	Executes program one source line at a time. Will step into calls.
next	Executes program one source line at a time. Will NOT step into calls.
print	Prints the value of variable or expression. Multiple items are separated by commas.

Table 1. Common TotalView line-mode commands

Command	Description
where	Lists functions currently active in program.
list	Lists 10 source lines.
quit	Quits debugger.

Scalable Debugging for Cray T3D Programs

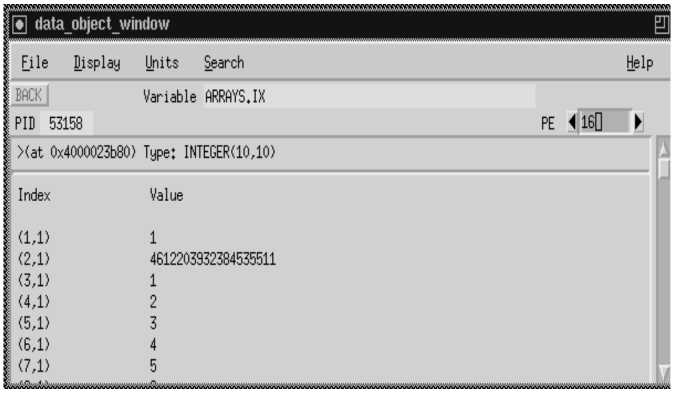
TotalView 2.0 extends TotalView's capabilities for Cray T3D programs using new methods for user-debugger interaction that promote greater understanding of parallel programs, making it easier to debug Cray T3D programs. It offers a scalable command set for unified control of many processors, improved ability to display Cray T3D private data across PEs and better process execution state information with navigational capabilities for many processors.

Scalable Commands

TotalView 2.0 lets users target a set of processors for debugging. This set of processors is called a *PSet*. A *PSet* is a collection of processors that is treated as a unit during debugging. Debugging commands will act on the processors defined in the current *PSet*. For example, if *PSet* is *All*, the *next* command will advance all processors to the next source line. If *PSet* is *Single* and current process element (PE) is 0, the *next* command will advance only PE 0 to the next source line. A *PSet* of *All* will be the default processor set on Cray T3D. Initially only two predefined process sets will be supported: *Single* and *All*.

Private Data

Cray T3D private data is replicated on all processors. Each executing processor owns it own copy of the data object. In TotalView 2.0, Cray T3D private data values across all processors can be displayed from one data window. The data window displays an input box and a process element (PE) number. The user simply enters a valid PE number to view the same private data object for that PE. Figure 3 shows an example of the PE input box. This interface is also available on the data array browser and can be used to pan private array data.

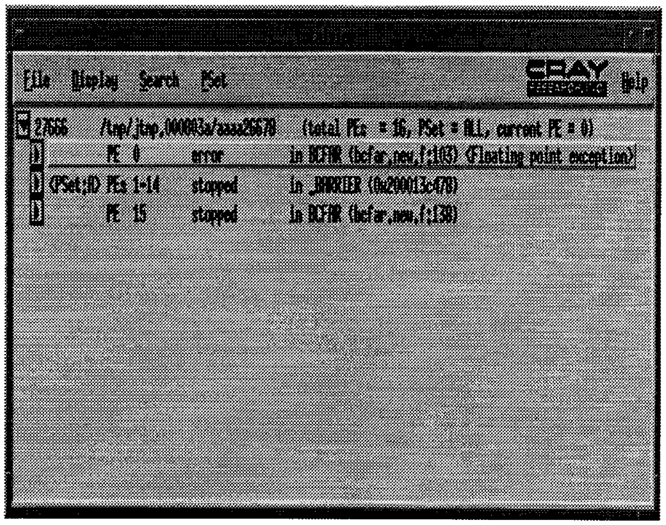


```
Variable ARRAYS.IX
PID 53158                                         PE 16
>(at 0x4000023b80) Type: INTEGER(10,10)

Index           Value

(1,1)           1
(2,1)           4612203932384535511
(3,1)           1
(4,1)           2
(5,1)           3
(6,1)           4
(7,1)           5
```

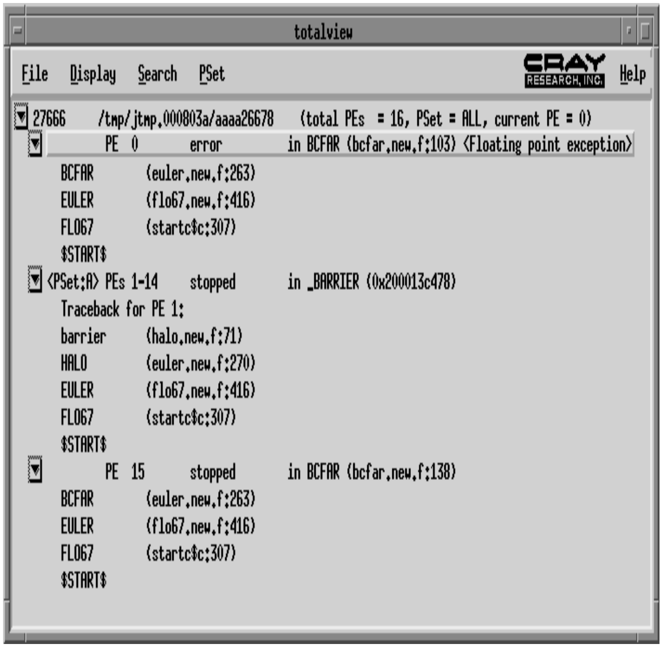
Figure 3: Data object window with PE input box



```
27666    /tmp/jtmp.000803a/aaaa26678    (total PEs = 16, PSet = ALL, current PE = 0)
             PE  0      error       in BCFAR (bcfar,new,f;103) <Floating point exception>
  <PSet;A> PEs 1-14     stopped     in _BARRIER (0x200013c478)
             PE  15     stopped     in BCFAR (bcfar,new,f;138)
```

Process Execution Status

TotalView 2.0 uses new ways to help users navigate among multiple processors in a Cray T3D program. The totalview window is the main Cray TotalView window. It is a system window that appears by default when you start the debugger. When the totalview window first appears for a parallel program only one entry is visible. This entry represents the user program. The entry displays the name, process ID, current status, and process set information. The user simply clicks on the arrow that



```
27666    /tmp/jtmp.000803a/aaaa26678    (total PEs = 16, PSet = ALL, current PE = 0)
             PE  0      error       in BCFAR (bcfar,new,f;103) <Floating point exception>
     BCFAR      (euler,new,f;263)
     EULER      (flo67,new,f;416)
     FLO67      (startc$c;307)
     $START$
  <PSet;A> PEs 1-14     stopped     in _BARRIER (0x200013c478)
     Traceback for PE 1:
     barrier    (halo,new,f;71)
     HALO       (euler,new,f;270)
     EULER      (flo67,new,f;416)
     FLO67      (startc$c;307)
     $START$
             PE  15     stopped     in BCFAR (bcfar,new,f;138)
     BCFAR      (euler,new,f;263)
     EULER      (flo67,new,f;416)
     FLO67      (startc$c;307)
     $START$
```

Figure 5: Process entry expanded to show traceback information

appears next to the entry name and the entry will be expanded to show individual entries for each process element (PE). Process entries that are exactly the same (that is, they have the same program location and status) will be noted in a single entry. The user can dive on any entry to have its information displayed in a process window. Expanding each process entry shows the current traceback information.

Release Plans

Cray TotalView 2.0 is scheduled for release in the fourth quarter of 1995 with CrayTools 2.0. TotalView revision 2.0.1 is planned for release in the first quarter of 1996. TotalView 2.0 and 2.0.1 will be released on all supported platforms including Cray Y-MP, C90, J90, T90, T90-IEEE, T3D, T3E and SPARC computer systems.