

Using DDT

Debugging programs with DDT

Peter Towers

HPC Systems Section

Peter.Towers@ecmwf.int

Allinea DDT

- **DDT is a very popular interactive debugger**
 - Developed by Allinea Software in the UK
 - Modern graphical user interface
 - Highly scalable to large numbers of tasks/threads
- **Cray provide a 2048 task licence**
 - No limit to threads per task
 - Support from Cray backed up by Allinea
- **Version 4.2.1 installed**
 - Cray have not yet released version 5 (Allinea Forge)
- **Also available on the latest ECMWF desktops**
 - Uses the same licence

DDT Features

- **Examine sources**
- **Set breakpoints**
 - Pause at a line of source
 - Conditional breakpoints supported
- **Examine variables, Fortran modules, stack**
- **Set tracepoints**
 - Output values of variables at a line without pausing
- **Set watchpoints**
 - Pause when a variable changes value

DDT Features

- **Step through execution line by line**
 - Step to the next line or into a function
 - Step over a line
 - Step out of a function
- **Catch signals**
 - Segmentation violations
 - Floating point exceptions
- **Track and debug memory usage**
 - Record all memory allocations
 - Find memory overwrites
- **Examine message queues**
-

Compilation

- **DDT uses the GNU Debugger (GDB) under the covers**
- **Compile with symbolic debug information turned on**
- **With CCE use**
 - g for debug info and no optimisation
 - G1 or -G2 for debug info with optimisation
- **With Intel use**
 - g for debug info

Launching DDT in batch

```
export DISPLAY=<your workstation>:0.0
```

```
module load ddt
```

```
ddt -n ? -mpiargs 'more aprun args' a.out
```

For example

```
ddt -n 4 -mpiargs '-N 4 -ss -cc cpu -d 6' a.out
```

Simple Job Script

```
#!/bin/ksh
#PBS -q np
#PBS -N ddtdemo
#PBS -j oe
#PBS -o job.out
#PBS -l EC_total_tasks=4
#PBS -l EC_threads_per_task=6
#PBS -l EC_hyperthreads=1
#PBS -l walltime=00:60:00

cd $HOME/Debug

module load ddt

export DISPLAY=juliet:0.0

export OMP_NUM_THREADS=6

ddt -n 4 -mpiargs '-ss -cc cpu -N 4 -d 6' ./hello_mpi
```

Live Demo Using IFS

- **TL159 running on 1 node**
 - Initially with 4 tasks x 6 threads
 - Then with 24 tasks x 1 thread
- **Latest RAPS14 benchmark code (CY41R2)**
- **Run in batch**
- **Selected screen shots from DDT follow in the presentation material**
- **Now for the live demo.....**

Before launching executable

The screenshot shows the 'Run' dialog box in Allinea DDT 4.2.1-36484. The dialog is titled 'Run <@ctmom2>' and contains the following configuration options:

- Application:** /perm/systems/sy6/RAPS14.dbg/bin/ifsMASTER (Details)
- Application:** /perm/systems/sy6/RAPS14.dbg/bin/ifsMASTER
- Arguments:** (empty)
- stdin file:** (empty)
- Working Directory:** (empty)
- MPI:** 4 processes, Cray X-Series (MPI/shmem/CAF) (Details)
 - Number of processes: 4
 - Processes per Node: 1
 - Implementation: Cray X-Series (MPI/shmem/CAF) (Change...)
 - aprun arguments: -ss -cc cpu -N 4 -d 6
- OpenMP:** 6 threads (Details)
 - Number of OpenMP threads: 6
- CUDA** (Details)
- Memory Debugging** (Details...)
- Submit to Queue** (Configure... Parameters...)
- Environment Variables:** none (Details)
- Plugins:** none (Details)

Buttons: Help, Run, Cancel

Licence Serial Number: 7929 ?

Support [Tutorials](#) [allinea.com](#)

Allinea DDT 4.2.1-36484

Paused after MPI_INIT

The screenshot shows the Allinea DDT 4.2.1-36484 interface. The main window displays the source code for `mpi_init_mod.F90`. The current line is 158, which is `LTHSAFEMPI = (IPROVIDED >= IREQUIRED)`. The code includes conditional compilation for MPI and OPS_COMPILE, and a warning message for POWER8 systems.

The **Locals** table shows the following variables:

Variable Name	Value
IPROVIDED	2
IREQUIRED	3
LTHSAFEMPI	.TRUE.

The **Stacks** table shows the following stack frames:

Processes	Function
4	master (master.F90:71)
4	dr_hook_default (yomhook.F90:46)
4	dr_hook_util (dr_hook_util.F90:60)
4	mpl_init (mpl_init_mod.F90:158)

After selecting cnt4.F90 and setting a breakpoint at line 505

The screenshot displays the Allinea DDT 4.2.1-36484 IDE interface. The main window shows the source code for the file `cnt4.F90`. A red circle indicates a breakpoint is set at line 505. The code includes several conditional statements and a `DO` loop. The `Stacks` panel at the bottom left shows the current stack frame: `mpl_init (mpl_init_mod.F90:158)`. The `Locals` panel is empty, and the `Evaluate` panel is also empty. The status bar at the bottom right shows "Ready".

```
491 ISTOP=NSTOP2-1
492 ENDF
493
494 IF ((IHISTS(ISTOP/NFRHIS) == 1.AND.MOD(ISTOP,NFRHIS) == 0).AND.I
495 ILAG=1
496 ELSE
497 ILAG=0
498 ENDF
499
500 IF (LSLAG.AND.(NCONF == 131.OR.NCONF == 401.OR.NCONF == 601.OR.N
501 LFINDVSEP=.TRUE.
502 ELSE
503 LFINDVSEP=.FALSE.
504 ENDF
505
506 DO JSTEP=NSTAR2, ISTOP
507
508 CALL USER_CLOCK(PTOTAL_CP=ZT1)
509
510 !* 3.1 Time filtering constraint based on digital filter
511
```

Hitting the breakpoint

The screenshot shows the Allinea DDT 4.2.1-36484 interface. The main window displays the Fortran source code for `cnt4.F90`. A breakpoint is set at line 506, which is `DO JSTEP=NSTAR2, ISTOP`. A dialog box titled "Program Stopped <@cctmom2>" is overlaid on the code, with the message: "Processes 0-3: Process stopped at breakpoint in cnt4 (cnt4.F90:506). Always show this window for user-defined breakpoints". The dialog box contains "Continue" and "Pause" buttons.

The code editor shows the following lines:

```
497   ILAG=0
498   ENDIF
499
500   IF (LSLAG.AND.(NCONF == 131.OR.NCONF == 401.OR.NCONF == 601.OR.I
501     LFINDVSEP=.TRUE.
502   ELSE
503     LFINDVSEP=.FALSE.
504   ENDIF
505
506   DO JSTEP=NSTAR2, ISTOP
507
508     CALL USER_CLOCK(PTO
509
510     !*   3.1   Time
511
512     IF (LJCDFI.AND.NCON
513       isub=1
514       ! Step should be
515     IF (NSUBDFI>1) TH
516       ipersub=NSTOP/N
517       istep=MOD(istep,ibersub)
```

The "Locals" panel on the right shows the following variables and values:

Variable Name	Value
ISTOP	12
JSTEP	0
NSTAR2	0

The "Input/Output" panel at the bottom left shows the following output:

```
Other: signal_dhook(SIGTERM=15): New handler installed
Other: signal_dhook(SIGXCPU=24): New handler installed
Other: signal_dhook(SIGSYS=31): New handler installed at
Other: MPL_BUFFER_METHOD: 2 128000000
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdats
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdats
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdats
```

Note: Allinea DDT can only send input to the aprun process with this MPI implementation

Type here ('Enter' to send): More

At the breakpoint

The screenshot displays the Alinea DDT 4.2.1-36484 debugger interface. The main window shows the source code of a Fortran program, with the current line of execution highlighted at line 506: `DO JSTEP=NSTAR2, ISTOP`. The code includes conditional logic for `LFINDVSEP` and a loop for time filtering. The `Locals` panel on the right shows the values of `ISTOP` (12), `JSTEP` (0), and `NSTAR2` (0). The `Input/Output` panel at the bottom left shows the output for the current process, including system messages and file reading operations. The `Evaluate` panel is currently empty.

Project Files

- cnt4.F90
- cnt4ad.F90
- cnt4tl.F90
- co2_tcmr.F90
- co2_tcmr_ad.F90
- co2_tcmr_tl.F90
- co2slicing.F90
- cobs.F90
- cobsad.F90
- cobsall.F90
- cobsallad.F90
- cobsalltl.F90
- cod_op.F90
- cod_op_tm5.F90
- cod_opad.F90
- cod_optl.F90

```
497   ILAG=0
498   ENDIF
499
500   IF (LSLAG.AND.(NCONF == 131.OR.NCONF == 401.OR.NCONF == 601.OR.NCONF == 801))
501     LFINDVSEP=.TRUE.
502   ELSE
503     LFINDVSEP=.FALSE.
504   ENDIF
505
506   DO JSTEP=NSTAR2, ISTOP
507
508     CALL USER_CLOCK(PTOTAL_CP=ZT1)
509
510     !*   3.1   Time filtering constraint based on digital filter
511
512     IF (LJCDFI.AND.NCONF==1) THEN
513       isub=1
514       ! Step should be local to subwindow
515       IF (NSUBDFI>1) THEN
516         ipersub=NSTOP/NSUBDFI
517         istep=MOD(istep,ipersub)
```

Variable Name	Value
ISTOP	12
JSTEP	0
NSTAR2	0

Input/Output

Output For Process: All

```
Other: signal_drhook (SIGTERM=15): New handler installed
Other: signal_drhook (SIGXCPU=24): New handler installed
Other: signal_drhook (SIGSYS=31): New handler installed
Other: MPL_BUFFER_METHOD: 2 12800000
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdata
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdata
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdata
```

Note: Alinea DDT can only send input to the aprun process with this MPI implementation

Type here ('Enter' to send): More

Ready

Setting a trace point at line 511 for JSTEP and ZT1 every 3rd step

The screenshot displays the Alinea DDT 4.2.1-36484 interface. The main window shows the source code for file `cnt4.F90`, with line 511 highlighted. The code at line 511 is:

```
511 !* 3.1 Time filtering constraint based on digi
```

The **Edit Tracepoint** dialog is open, showing the following configuration:

- Location:**
 - File: `/usr/snx11057/perm/systems/sy6/RAPS14.dbg/src/ifs/control/cnt4.F90`
 - Line Number: `511`
- Applies To:**
 - Process Group: `All`
 - Process: `All`
 - Thread: `All`
- Tracepoint Output:**
 - Tracepoint Name: (empty)
 - Variables Logged: `JSTEP`, `ZT1`
- HR Limits:**
 - Start on the n-th pass: `1`
 - Trigger every n-th pass: `3`
 - Log at most n times: `Forever`
- Condition:** (empty)
- Language:** `Auto`

The **Input/Output** window shows the following output for the process:

```
Other: signal_drhook(SIGTERM=15): New handler installed at
Other: signal_drhook(SIGXCPU=24): New handler installed at
Other: signal_drhook(SIGSYS=31): New handler installed at
Other: MPI_BUFFER_METHOD: 2 128000000
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdata
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdata
Other: Reading /perm/systems/sy6/RAPS14.dbg/data/ifsdata
```

Note: Alinea DDT can only send input to the aprun process with this MPI implementation

Type here ('Enter' to send):

Setting a conditional breakpoint at line 507 for step 8

The screenshot shows the Allinea DDT 4.2.1-36484 interface. The 'Edit Breakpoint' dialog box is open, showing the following configuration:

- Location: Line
- File: /usr/snx11057/perm/systems/sy6/RAPS14.dbg/src/ifs/control/cnt4.F90
- Line Number: 507
- Applies To: Process Group: All, Process: All, Thread: All
- Hit Limits: Start on the n-th pass: 0, Trigger every n-th pass: 1, Stop after n hits: Forever
- Condition: Condition: JSTEP.eq.8
- Language: Auto

The background shows the project files tree with 'cnt4.F90' selected, and the code editor displaying the following code snippet:

```
503 LFINDV
504 ENDIF
505
506 DO JSTEP
507
508 CALL U
509
510 !*
511
512 IF (L
513 isub
514 ! St
515 IF (
516 ip
517 is
518 is
519 !
520 IF
521
522
523 EN
```

Trace point output after 8 steps

The screenshot shows the Allinea DDT 4.2.1-36484 interface. The main window displays the source code for 'cnt4.F90' with line 508 highlighted: `CALL USER_CLOCK(PTOTAL CP=ZT1)`. The 'Locals' panel shows the variable 'ZT1' with a value of `-342.86542800000001`. The 'Tracepoint Output' panel shows a table of logged values for JSTEP and ZT1 at different steps.

Tracepoint	Processes	Values logged
cnt4 (cnt4.F90:511)	4, ranks 0-3	JSTEP: <code>1</code> ZT1: <code>-342.86542800000001</code> from 7.2e+01 to 7.3e+01
cnt4 (cnt4.F90:511)	4, ranks 0-3	JSTEP: <code>4</code> ZT1: <code>-342.86542800000001</code> from 2.12e+02 to 2.13e+02
cnt4 (cnt4.F90:511)	4, ranks 0-3	JSTEP: <code>7</code> ZT1: <code>-342.86542800000001</code> from 3.43e+02 to 3.44e+02

Running a 24 x 1 configuration

- **IFS runs until it gets a Floating Point Exception**
- **Caused by a divide by zero**
- **The physics time slice has gone to zero**
- **Will use a watch point to find out where**

Hitting the divide by zero

The screenshot shows the Alinea DDT 4.2.1-36484 interface. The main window displays the source code for 'qnegat.F90'. The error message is:

```
Process 4:  
Process stopped in qnegat (qnegat.F90.89) with signal SIGFPE (Arithmetic exception).  
Reason/Origin: floating point divide by zero  
Your program will probably be terminated if you continue.  
You can use the stack controls to see what the process was doing at the time.  
 Always show this window for signals
```

The stack window shows the following variables:

Variable Name	Value
ZCONS1	-9.1702764791777932e-05
ZTMST	0

The source code snippet shows the following lines:

```
80 !* COMPUTATIONAL CONSTANTS.  
81 !  
82 !-----  
83 ! ZTMST=TWODT  
84 ! IF (NSTEP.EQ.NSTART) ZTMST=0.5*TWODT  
85 IF (LHOOK) CALL DR_HOOK('QNEGAT', 0, ZHOOK_HANDLE)  
86 ASSOCIATE (REPQMI => YRECND%REPQMI)  
87 ZTMST=PTSPHY  
88  
89 ZCONS1=1.0 J  
90  
91 !INITIALIZAT  
92 IF (.NOT. PR  
93  
94 !***  
95  
96 !-----  
97  
98 !* 2  
99 !  
100
```

The 'Input/Output' window shows the following output:

```
Other: *** IVECTOR= 1  
Other: *** LPRRPROC= F  
Other: *** LWCOUNORMS= F  
Other: *** LSMSSIG_WAM= F  
Other: =====  
aprun: 11:36:14 STEP 1 H= 1:00 +CPU= 6.376  
aprun: 11:36:20 STEP 2 H= 2:00 +CPU= 5.812
```

The 'Evaluate' window is empty.

After reaching a breakpoint in callpar.F90 and setting a watch point on tsphy

The screenshot shows the Alinea DDT 4.2.1-38484 debugger interface. The main window displays the source code for 'callpar.F90' with a red circle on line 422. The 'Locals' panel on the right shows the variable 'tsphy' with a value of 3600. The 'Watchpoints' panel at the bottom left shows a watchpoint set on 'tsphy'.

```
413 & YSP_RR=>YRSURF%YSP_RR, YSP_SG=>YRSURF%YSP_SG)
414
415 ! -----
416
417 !*      0.      INITIALIZATION
418 !
419 !      0.1    Initialization of constants
420
421 ! this should be moved to setup with some L_CLOUDITER key in phys
422 LL_CLOUDITER=.TRUE. .AND. (.NOT. LPHYLIN).AND. (.NOT. YREPHY%LSP
423
424 LLSLPHY = LSLPHY.AND.LSLAG
425 LLRAINID = .FALSE. ! key reserved for 1D var rain
426
427 ! constants
428 ZRG=1.0_JPRB/RG
429 ZRCPD=1.0_JPRB/RCPD
430
431 ! Some sort of security (I wonder why this is not in the setup)
432 IF (LEPCLD .AND. .NOT. (YA%LACTIVE.AND.YL%LACTIVE.AND.YI%LACTIVE
433 CALL ABORT('CALLPAR: How did we get here?')
```

Variable Name	Value
!\$7	0x8bce0 <yoePHY_>
!\$79	65927
!\$8	0x8bbb20 <yoeaeratm_>
!\$89	65928
!\$9	0x8bd0c0 <yoeRad_>
!\$96	65748
!\$99	65929
tendency_cml	
tendency_dyn	
tendency_loc	
tendency_tmp	
tendency_vdf	
tsphy	3600
ygfl	(numfids = 17, nders = 0, numspflds = 0, numgpflds = 17, numfids9 =
yrchem	
yrCumfs	(lCumfs = .FALSE., lregcv = .FALSE.)
yrdebu	(nproc = 1, nproc = 0, nproc = 0, nproc = 0, nproc = 1, nproc = 0, nproc =

Processes	Scope	Expression
<input checked="" type="checkbox"/>	All	#0 callpar tsphy

After the watch point triggers in sitend.F90

The screenshot displays the AAllinea DDT 4.2.1-36484 interface. The main window shows the Fortran source code for `sitend.F90`. A watchpoint has been triggered at line 415, where the variable `tsphy` is updated. A dialog box titled "Program Stopped <@cctmom2>" is overlaid on the code, providing details about the stop:

- Process 4:
- Process stopped at watchpoint "tsphy" in sitend (sitend.F90:415).
- Old value: 3600
- New value: 0
- Always show this window for watchpoints
- Buttons: Continue, Pause

The background interface includes:

- Project Files:** A tree view on the left showing various Fortran modules like `sitnu.F90`, `sitnuad.F90`, etc., with `sitend.F90` selected.
- Code Editor:** Shows lines 398-417 of `sitend.F90`. Line 415 is highlighted: `IF (LHOOK) CALL DR`.
- Locals Panel:** A table on the right showing the current state of local variables:

Variable Name	Value
zepsilon	2.2204460492503131e-14
zfac	1.0454158174220698
zfaccc	0.28346868942621073
zhook_handle	3.2430270181003537e-315
zpp	([1] = 94379.106594719313, ...)
zpt	([1] = 264.15581262010329, ...)
([1] = 0, ...)	([1] = 0, ...)
([1] = 0.0016206026841087756, ...)	([1] = 0.0016206026841087756, ...)
([1] = 0.0028522486915119531	([1] = 0.0028522486915119531
([1] = 0.10197162129779283	([1] = 0.10197162129779283
([1] = 0.017217980924555294, ...)	([1] = 0.017217980924555294, ...)
([1] = 0.0017161058851406594, ...)	([1] = 0.0017161058851406594, ...)
([1] = 262.82154429644334, ...)	([1] = 262.82154429644334, ...)
([1] = 1.0909490209890669, ...)	([1] = 1.0909490209890669, ...)
([1] = 0.017217980924555294, ...)	([1] = 0.017217980924555294, ...)
([1] = 1.0909490209890669, ...)	([1] = 1.0909490209890669, ...)
- Input/Output Panel:** Shows the output of the `aprun` process, including system parameters like `IVECTOR= 1`, `LPRKPROC= F`, and performance metrics for two steps.

Line 411 did it!

The screenshot shows the Allinea DDT 4.2.1-36484-@octm2m2015 interface. The main window displays Fortran code from the file `mpl_init_mod.F90`. The code includes conditional logic for setting `PTENA` and `PTENO3` based on `YA%LPHY` and `Y03%LPHY`. Line 411 is highlighted, showing the statement `if(myproc.eq.5.and.NSTEP.eq.3) tsphy=0.0`. The `Locals` window on the right lists variables such as `zepsilon`, `zfac`, `zfaccc`, `zhook_handle`, `zpp`, `zpt`, `zqad`, `zqold`, `zqp1env`, `zrg_r`, `zsuba`, `zsubq`, `zsubt`, `zzfac`, `zzsuba`, and `zzzfac`. The `Output` window at the bottom left shows the following output for process 1:

```
Other: *** IVECTOR= 1
Other: *** LPRRPROC= F
Other: *** LWCOUNORMS= F
Other: *** LSMSSIG_WAM= F
Other: =====
aprun: 12:41:03 STEP 1 H= 1:00 +CPU= 6.364
aprun: 12:41:08 STEP 2 H= 2:00 +CPU= 5.848
```

The `Evaluate` window is currently empty. The status bar at the bottom right indicates "23 processes playing".

Allinea Map

- **A product that complements DDT**
- **A simple code profiler for MPI applications**
- **Version 5 will support OpenMP as well**
- **Same source code browser as DDT**
- **Find hot spots in the application**
- **Available on the ECMWF systems**
- **See the DDT/MAP userguide for more information**
 - `/opt/cray/ddt/4.2.1.2_36484/doc/userguide.pdf`

Very Simple Example

```
tar -xzf ~trx/exercises/debug.tgz
```

```
cd Debug
```

```
compit
```

```
edit job.ddt and set DISPLAY
```

```
qsub job.ddt
```

Try setting

- a breakpoint at line 26
- a watch point on nthreads
- a conditional trace point on line 50 for nstep 50

Find the error output and trace output when it fails