



TOTALVIEW DEBUGGING ON XEON PHI

TotalView

- **Multiple compilers and platforms**
 - C/C++, F77/95, UPC
 - Linux Clusters, Cray XC, IBM BG/Q, OS X etc.
 - CUDA GPUs, Intel Xeon Phi
 - Multi-threaded, Parallel (MPI, PVM)
- **Memory debugging *)**
- **Reverse debugging *)**
- **Easy to use GUI and powerful CLI**
 - Commonality across all supported platforms
- **Remote debugging**
- **Unattended batch debugging**

*) These features currently on CPU only

X86-64 Linux Compatibility



64-Bit x86-64 Linux								
Operating Systems	Environment/Compiler	Product	Versions				Notes	
Novell Open SuSE 11.1	C and C++	GNU GCC	3.4	4.0	4.1	4.2	TotalView Suite Build and Certification Platform SuSE Linux Enterprise Server	
Novell SuSE Linux Enterprise Server 10, 11, and 11.1		GNU GCC	4.3	4.4	4.5	4.6		
Red Hat Enterprise Linux 4, 5, and 6	FORTRAN 77 and Fortran 90	Intel C/C++ Compiler for Linux	11	11.1	12	12.1	ReplayEngine Support ReplayEngine supports the OS and compiler variants listed above for 64-Bit x86-64 Linux. ReplayEngine supports the IP transport mechanism in the following MPI versions: <ul style="list-style-type: none"> Argonne MPICH, version 1.1, 1.2, 1.3, and 1.4 Argonne MPICH2 1.0.7, 1.1, and 1.2 Open MPI 1.2.8, 1.3.2, and 1.4.2 Intel MPI 3.0, 4.0, and 4.0.3 SGI MPT 1.26 and 1.27 Cray XT-MPT 2.0 OSU MVAPICH 0.9.9, 1.1, and 1.2 OSU MVAPICH2 1.4.1, 1.5, 1.8, and 1.7 IBM Platform MPI 8.3 ReplayEngine supports native communication over Infiniband using either the IBverbs or the QLogic PSM transport layers in the following MPI versions: <ul style="list-style-type: none"> Open MPI 1.4.2 OSU MVAPICH 1.2 OSU MVAPICH2 1.5, 1.8, and 1.7 Intel MPI 4.0 and 4.0.3 IBM Platform MPI 8.3 In some circumstances, prerequisites exist for using ReplayEngine with Infiniband MPIs. See the TotalView Users Guide section "Using ReplayEngine with Infiniband MPIs".	
Red Hat Fedora 12, 13, 14, and 15		Pathscale EKO	3.1	3.2				
SGI Altix SLES 10 SP1 and ProPack 5 SP3		PGI Workstation	10.1	10.6	11.2	11.9		
Ubuntu Linux 8, 9, 10, and 11.04				12.1				
Certification platforms: SuSE Linux Enterprise Server 11.1 RedHat Enterprise Server 6			Absoft Pro	9.0	10.0			
			GNU GCC	3.4 (FORTRAN 77 only)				
			GNU gfortran	4.1.2 - 4.2 (RH ES 5u2)				
				4.3	4.4	4.5		4.6
			Intel Fortran Compiler for Linux	11	11.1	12		12.1
			PGI Workstation	10.1	10.6	11.2		11.9
			12.1					
		Pathscale EKO	3.1	3.2				
		Lahey Linux 64-bit and Fortran Pro 64-bit	6.2	8.0				
		Sun Studio	12					
	MPI	Argonne MPICH	1.2.7					
		Argonne MPICH2	1.1	1.2	1.3	1.4		
		GNU SLURM	1.2					
		Intel MPI	3.0	4.0	4.0.3			
		Open-MPI.org Open MPI	1.2.8	1.3	1.3.2	1.4.2		
				1.5				
			OSU MVAPICH	1.2				
			OSU MVAPICH2	1.4.1	1.5	1.6	1.7	
			SGI MPT	1.2.6	2.0			
			SGI Propack	5 SP3	6			
		Bulix MPI	1.1.3					
		IBM Platform MPI	8.3					
	OpenMP C/C++	Intel C/C++ Compiler for Linux	11	11.1	12	12.1		
		PGI Workstation	10.1	10.6	11.2	11.9		
				12.1				
		Sun Studio	12					
		GNU GCC	4.1.2	4.2.0	4.2.3	4.4		
			4.5	4.6				
	Open MP FORTRAN 77 and Fortran 90	GNU gfortran	4.1.2 - 4.2 (RH ES 5u2)					
			4.4	4.5	4.6			
		Intel Fortran Compiler for Linux	11	11.1	12	12.1		
		PGI Workstation	10.1	10.6				

Supported platforms for v. 8.11

Features on Xeon Phi

- Improved support in version 8.12
 - Currently in beta testing (8.12T)
- Native applications
 - Remote debugging from host
- Offload applications
 - Full visibility of both host and coprocessor threads
 - Asynchronous thread control on both host and coprocessor
- MPI support

Simple Example



- Compile the program with the **-g** flag

```
$ icc -openmp -g -mmic ./omphello.c -o omphello
```

- Load totalview module

```
$ module load totalview
```

- Load the program in the TotalView GUI

- Remote debugging parameter **-r mic0**
- Executable name **./omphello**

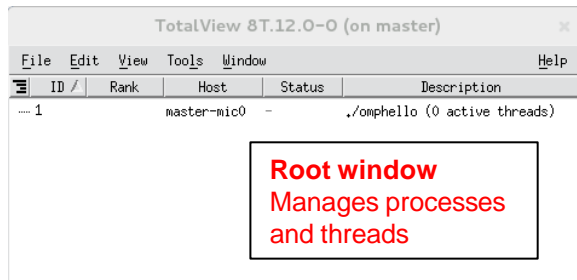
```
$ totalview -r mic0 ./omphello
```

Note: X Windows forwarding must be enabled!

Initial GUI view

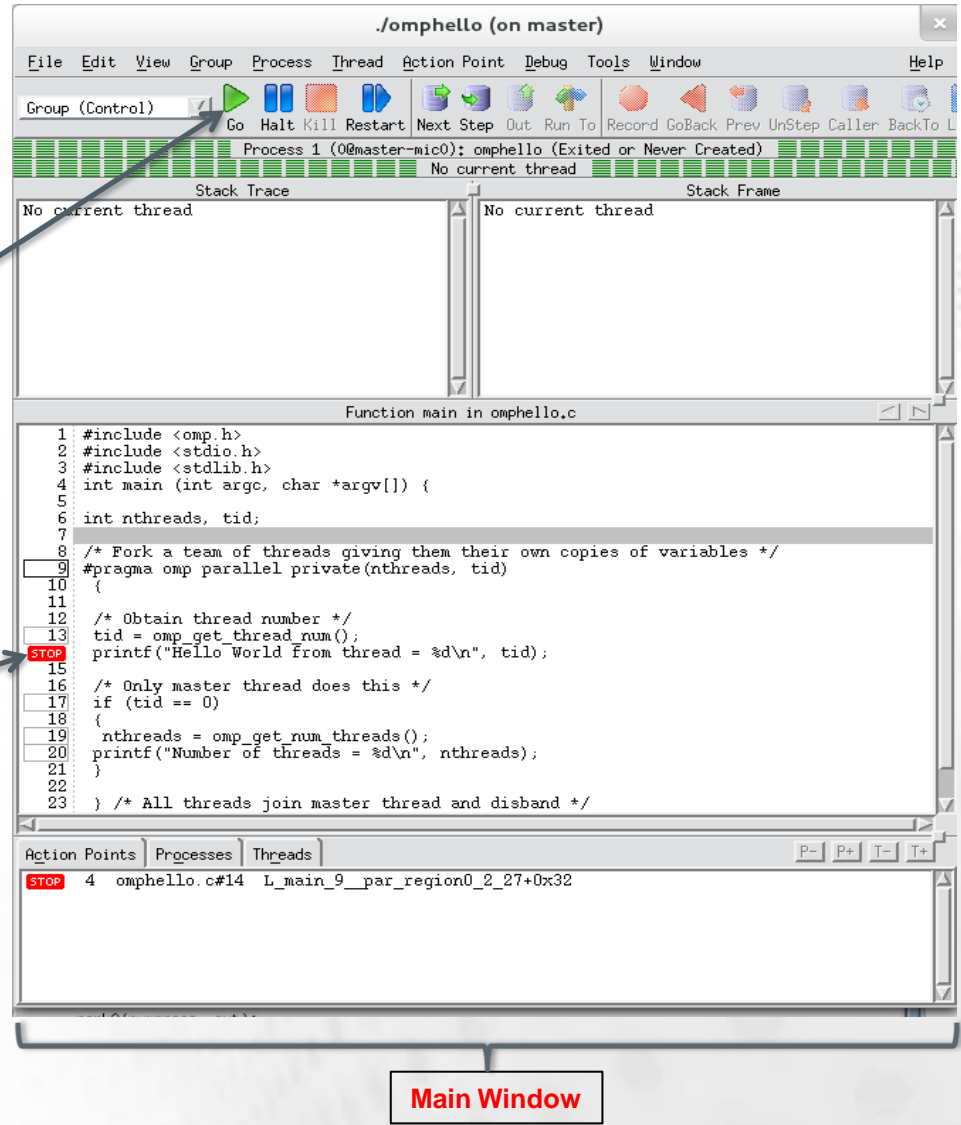
These buttons control process execution. Click **Go** to start!

Set breakpoints by clicking on the line #



Root window
Manages processes and threads

ID /	Rank	Host	Status	Description
1		master-mic0	-	./omphello (0 active threads)



Main Window

Function main in omphello.c

```

1  #include <omp.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  int main (int argc, char *argv[]) {
5
6  int nthreads, tid;
7
8  /* Fork a team of threads giving them their own copies of variables */
9  #pragma omp parallel private(nthreads, tid)
10 {
11
12  /* Obtain thread number */
13  tid = omp_get_thread_num();
14  printf("Hello World from thread = %d\n", tid);
15
16  /* Only master thread does this */
17  if (tid == 0)
18  {
19      nthreads = omp_get_num_threads();
20      printf("Number of threads = %d\n", nthreads);
21  }
22
23  } /* All threads join master thread and disband */

```

Action Points | Processes | Threads

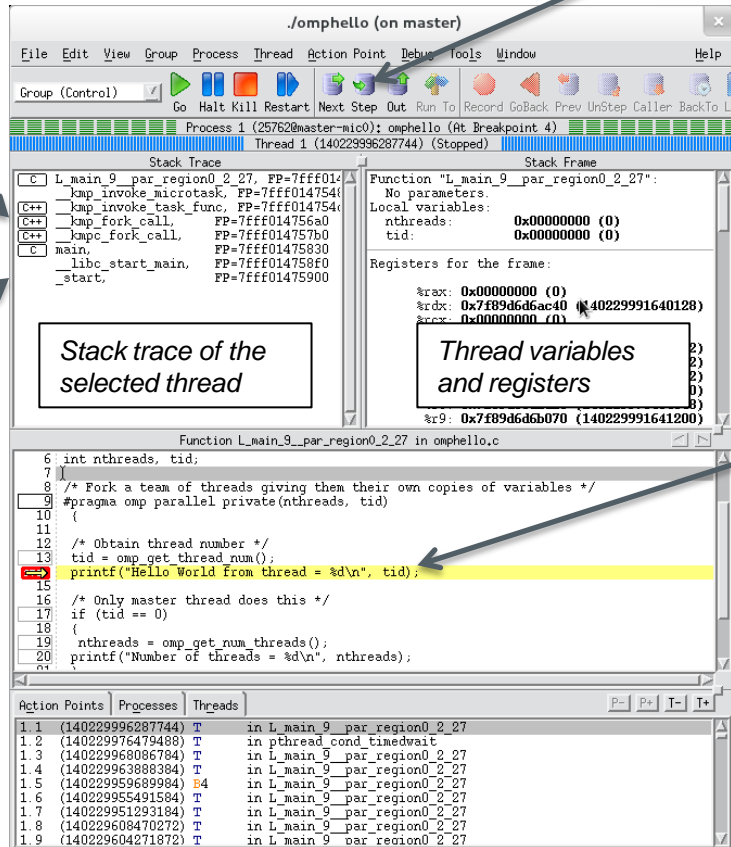
STOP	4	omphello.c#14	L_main_9_par_region0_2_27+0x32
------	---	---------------	--------------------------------

Analysis at the Breakpoint

Several levels of special Intel OpenMP (KMP) calls

Click on the callsite to move to that level of stack

Advance in the program



Stack trace of the selected thread

```

L_main_9_par_region0_2_27, FP=7fff0147548
_kmp_invoke_microtask, FP=7fff0147548
_kmp_invoke_task_func, FP=7fff0147548
_kmp_fork_call, FP=7fff014756a0
_kmpc_fork_call, FP=7fff014757b0
main, FP=7fff01475830
_libc_start_main, FP=7fff014758f0
_start, FP=7fff01475900
    
```

Thread variables and registers

```

Local variables:
nthreads: 0x00000000 (0)
tid: 0x00000000 (0)

Registers for the frame:
%rax: 0x00000000 (0)
%rbx: 0x7f89d6d6ac40 (140229991640128)
%rcx: 0x00000000 (0)
%rdi: 0x00000000 (0)
%rsi: 0x00000000 (0)
%r15: 0x7f89d6d6b070 (140229991641200)
    
```

Source code:

```

6 int nthreads, tid;
7
8 /* Fork a team of threads giving them their own copies of variables */
9 #pragma omp parallel private(nthreads, tid)
10 {
11
12 /* Obtain thread number */
13 tid = omp_get_thread_num();
14 printf("Hello World from thread = %d\n", tid);
15
16 /* Only master thread does this */
17 if (tid == 0)
18 {
19 nthreads = omp_get_num_threads();
20 printf("Number of threads = %d\n", nthreads);
21 }
22 }
    
```

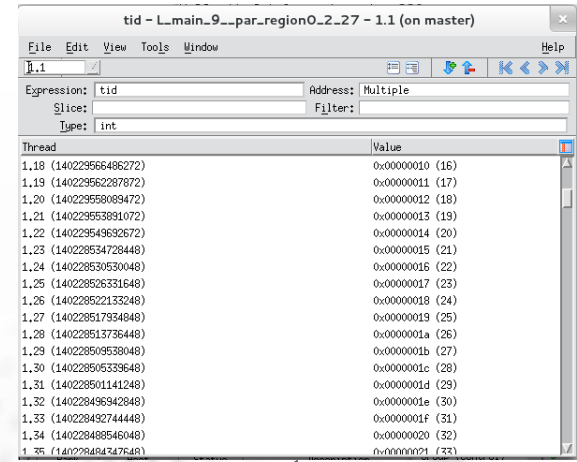
Action Points:

ID	Rank	Host	Status	Description
1.1		master-mic0	T	in L_main_9_par_region0_2_27
1.2		master-mic0	T	in pthread_cond_timedwait
1.3		master-mic0	T	in L_main_9_par_region0_2_27
1.4		master-mic0	T	in L_main_9_par_region0_2_27
1.5		master-mic0	B4	in L_main_9_par_region0_2_27
1.6		master-mic0	T	in L_main_9_par_region0_2_27
1.7		master-mic0	T	in L_main_9_par_region0_2_27
1.8		master-mic0	T	in L_main_9_par_region0_2_27
1.9		master-mic0	T	in L_main_9_par_region0_2_27

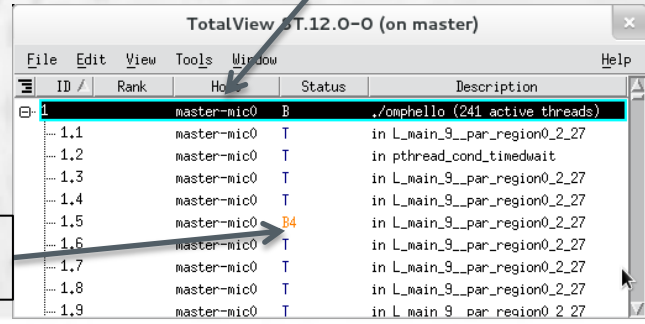
B* indicates which thread hit the breakpoint

Click on variables to compare values across threads

Select the thread to analyze in the root window



Thread	Value
1.18	0x00000010 (16)
1.19	0x00000011 (17)
1.20	0x00000012 (18)
1.21	0x00000013 (19)
1.22	0x00000014 (20)
1.23	0x00000015 (21)
1.24	0x00000016 (22)
1.25	0x00000017 (23)
1.26	0x00000018 (24)
1.27	0x00000019 (25)
1.28	0x0000001a (26)
1.29	0x0000001b (27)
1.30	0x0000001c (28)
1.31	0x0000001d (29)
1.32	0x0000001e (30)
1.33	0x0000001f (31)
1.34	0x00000020 (32)
1.35	0x00000021 (33)



ID	Rank	Host	Status	Description
1.1		master-mic0	T	in L_main_9_par_region0_2_27
1.2		master-mic0	T	in pthread_cond_timedwait
1.3		master-mic0	T	in L_main_9_par_region0_2_27
1.4		master-mic0	T	in L_main_9_par_region0_2_27
1.5		master-mic0	T	in L_main_9_par_region0_2_27
1.6		master-mic0	B4	in L_main_9_par_region0_2_27
1.7		master-mic0	T	in L_main_9_par_region0_2_27
1.8		master-mic0	T	in L_main_9_par_region0_2_27
1.9		master-mic0	T	in L_main_9_par_region0_2_27

Offload applications

- Compile and load the Totalview module
- Load the program in the TotalView GUI
 - No need for remote execution parameter
 - Executable name **./omphello_offload**

```
$ totalview ./omphello_offload
```

- Set up the debugging in the GUI
 - Add breakpoints etc.
 - Press "Go" to launch execution

Debugging an Offload Application

Select host or coprocessor threads from root window

ID	Rank	Host	Status	Description
1		<local>	T	./offload_hello (4 active threads)
1.1		<local>	T	in pthread_mutex_lock
1.2		<local>	T	in __poll
1.3		<local>	T	in __poll
1.4		<local>	T	in __poll
2	0	master-mic0	B	offload_hello.offload,0 (240 active threads)
2.1	0	master-mic0	T	in sem_wait
2.2	0	master-mic0	T	in L_main_31__par_loop1_2_52
2.3	0	master-mic0	T	in __poll
2.4	0	master-mic0	T	in pthread_cond_wait
2.5	0	master-mic0	T	in pthread_cond_timedwait
2.6	0	master-mic0	B1	in L_main_31__par_loop1_2_52
2.7	0	master-mic0	T	in L_main_31__par_loop1_2_52
2.8	0	master-mic0	T	in L_main_31__par_loop1_2_52

Several special control threads

Offload section runs in special program called *offload_main*

Process 1 (17420): offload_hello (Stopped)

Thread 1 (140616065304352) (Stopped)

Stack Trace

```
pthread_mutex_lock, FP=7fff062e7460
..._entHandler10AddWaiters@coievent, 1
..._COIEventWait@PKR@coievent@hPjS2, 1
COIEventWait, FP=7fff062e7580
..._loadDescriptor14offload_finishEv, 1
..._arDesc2iPPriS6_TAG_IGBASE_0_394, 1
offload_offload, FP=7fff062e7740
main, FP=7fff062e78f0
..._libc_start_main, FP=7fff062e79b0
_start, FP=7fff062e79c0
```

Stack Frame

Function "main":
No parameters.
Local variables:
nThreads: 0x00000001 (1)
i: 0x2c689958 (745052504)

Registers for the frame:

```
%rax: 0x00000000 (0)
%rdx: 0x00000000 (0)
%rcx: 0x00000001 (1)
%rbx: 0x00000002 (2)
%rsi: 0x00000000 (0)
%rdi: 0x00000001 (1)
%rbp: 0x7fff062e78f0 (140733297096944)
%rsp: 0x7fff062e7750 (140733297096528)
%r8: 0x7fe3ba525fc0 (140616060264384)
..._0x00000000_0_717400
```

Function main in offload_hello.c

```
25
26 int main()
27 {
28     const int nThreads=1;
29     int i;
30     #pragma offload target(mic)
31     #pragma omp parallel for
32     for(i=0; i < nThreads; i++) hello(5);
33     return 0;
34 }
```

Action Points Processes Threads

1 2

Device 0: offload_hello.offload,0 (Breakpoint 1)

Thread 6 (140495997843200): offload_main (At Breakpoint 1)

Stack Trace

```
L_main_31__par_loop1_2_52, FP=7fc7c601
..._kmp_invoke_microtask, FP=7fc7c60b29f0
..._kmp_invoke_task_func, FP=7fc7c60b29f0
..._kmp_launch_thread, FP=7fc7c60b2a30
..._kmp_test_then_and64, FP=7fc7c60b2de0
_start_thread, FP=7fc7c60b2ef0
..._clone, FP=7fc7c60b2ef8
```

Stack Frame

Function "L_main_31__par_loop1_2_52":
No parameters.
Local variables:
nThreads: 0x00000001 (1)
i: 0x00000000 (0)

Registers for the frame:

```
%rax: 0x00000001 (1)
%rdx: 0x7fc7c72eaa68 (140496016943720)
%rcx: 0x00000001 (1)
%rbx: 0x7fc7c64ea2ec (140496002261740)
%rsi: 0x7fc7c60b2998 (140495997839768)
%rdi: 0x7fc7c60b29a0 (140495997839776)
%cbp: 0x7fc7c60b2970 (140495997839728)
%rsp: 0x7fc7c60b2820 (140495997839392)
%r8: 0x7fc7c628ed20 (14049599790368)
..._0x7fc7c60b2820_140496074104944
```

Function L_main_31__par_loop1_2_52 in offload_hello.c

```
29 int i;
30 #pragma offload target(mic)
31 #pragma omp parallel for
32 for(i=0; i < nThreads; i++) hello(5);
33 return 0;
34 }
```

Action Points Processes Threads

2.5 (140496025339648) T in pthread_cond_timedwait

2.6 (140495997843200) B1 in L_main_31__par_loop1_2_52

2.7 (140495993644800) T in L_main_31__par_loop1_2_52

2.8 (140495989446400) T in L_main_31__par_loop1_2_52

2.9 (140495985248000) T in L_main_31__par_loop1_2_52

2.10 (140495981049600) T in L_main_31__par_loop1_2_52

2.11 (140495976851200) T in L_main_31__par_loop1_2_52

2.12 (140495972652800) T in L_main_31__par_loop1_2_52

2.13 (140495968454400) T in L_main_31__par_loop1_2_52

2.14 (140495359571712) T in L_main_31__par_loop1_2_52

Native MPI Applications

➤ Running on a single MIC

```
$ totalview -args mpiexec -np 4 -host mic0 ./mpihello
```

➤ Running on multiple MICs

- If you "own" the system and can define nodes explicitly

```
$ totalview -args mpiexec -np 4 -host \  
node02-mic0,node03-mic0 ./mpihello
```

- In the CSC cluster you can use it via SLURM
 - Give `mpirun-mic` the `--tv` flag

```
$ srun -N 2 mpirun-mic --tv -m ./mpihello
```

More information

- www.roguewave.com
 - Request an Evaluation License
 - Documentation
 - User Forums

- TotalView 8.12 Beta
 - Contact Nikolay Piskun
 - nikolay@roguewave.com

Extra: Gritty details

Single server launch (default)

```
totalview -args mpiexec -np 240 -hosts host1-mic0,host1-mic1,host2-  
mic0,host2-mic1 ./tx_basic_mpi  
set env TVDSVRLAUNCHCOMMAND=<your ssh command to card> (ssh,micssh)  
Set TV::server_launch_string preference
```

MIC Native Launch

```
totalview -mmic -args mpiexec -np 240 -hosts host1-mic0,host1-mic1,host2-  
mic0,host2-mic1 ./tx_basic_mp  
Set: dset TV::mic_native_server_launch_string {  
    ssh -n %R "/bin/rm -f /tmp/tvdsvrmain%K"; //1  
    scp %B/tvdsvrmain%K %R:/tmp/tvdsvrmain_mic; //2  
    ssh -n %R -n "/tmp/tvdsvrmain%K -callback %L -set_pw %P\ -verbosity  
%V %F" //3 }  
}
```

Removes your previous tvdsvrmain_mic

Copies it from the installation directory to the /tmp/ directory on the coprocessor

Starts the server on the Xeon Phi coprocessor.