The P*s* and Q*s* of Parallel Debugging

ALCF Early Science Program Workshop
March 19, 2012

Ray Loy
Applications Performance Engineering
ALCF

Outline

- bgp_stack
- coreprocessor
- gdb
- Allinea DDT
- TotalView

Interpreting your job's .error

```
…
<Jan 23 06:54:21.008467> FE_MPI (Info) : Starting job 535016
<Jan 23 06:54:21.085232> FE_MPI (Info) : Waiting for job to terminate
<Jan 23 06:54:23.515642> BE_MPI (Info) : IO - Threads initialized
<Jan 23 06:54:23.537559> BE_MPI (Info) : I/O input runner thread terminated
<Jan 23 06:54:33.589347> BE_MPI (Info) : I/O output runner thread terminated
<Jan 23 06:54:33.644169> BE_MPI (Info) : Job 535016 switched to state TERMINATED
 ('T')
<Jan 23 06:54:33.644228> BE_MPI (Info) : Job successfully terminated - TERMINATE
D ('T')
<Jan 23 06:54:33.719443> FE_MPI (Info) : Job terminated normally
<Jan 23 06:54:33.719541> FE_MPI (Info) : exit status = (139)
<Jan 23 06:54:33.719747> BE_MPI (Info) : Starting cleanup sequence
<Jan 23 06:54:33.719788> BE_MPI (Info) : cleanupDatabase() - job already termina
ted / hasn't been added
<Jan 23 06:54:33.750097> BE_MPI (ERROR): The error message in the job record is
as follows:
<Jan 23 06:54:33.750147> BE_MPI (ERROR):   "killed with signal 11"
<Jan 23 06:54:33.762069> BE_MPI (Info) : Destroying partition ANL-R00-M1-N04-64
<Jan 23 06:55:08.913586> BE_MPI (Info) : Partition ANL-R00-M1-N04-64 switched to
 state FREE ('F')
<Jan 23 06:55:09.162052> FE_MPI (Info) : ==   FE completed   ==
<Jan 23 06:55:09.162126> FE_MPI (Info) : == Exit status:   0 ==
<Jan 23 06:55:09.162203> SCHED_IF (Info) : mpirun result code: 0
<Jan 23 06:55:09.164395> SCHED_IF (Info) : job result code: 139
<Jan 23 06:55:09.184948> SCHED_IF (Info) : boot failure: False
<Jan 23 06:55:09.314332> SCHED_IF (Info) : resources associated with partition A
NL-R00-M1-N04-64 have been released
<Jan 23 06:55:09.334075> SCHED_IF (Info) : scheduler library unloaded
```

Lightweight core files

- Look for files
  - core.0, core.1, etc.
- Lightweight core files
  - One for each rank that failed
  - Contain stack backtrace in *address* form
  - Decode to symbolic (useful!) form
- Environment settings for core files
  - http://www.alcf.anl.gov/resource-guides/debugging ("Core file settings")

Lightweight Core File Example (BG/P; Q similar)

```
+++PARALLEL TOOLS CONSORTIUM LIGHTWEIGHT COREFILE FORMAT version 1.0
+++LCB 1.0
Program: /gpfs/home/rloy/public/winter-workshop-2012/hellompi
Job ID : 535016
Personality:
  XYZT coordinates : 0,0,0,0
  MPI Rank       : 0
  DDR Size (MB)    : 2048
  Mode           : SMP
+++ID TGID 100, Core 0, Thread 1 State 40000000, Sched: 48000000
General Purpose Registers:
 r00=00000078 r01=02100ee0 r02=021087a0 r03=00000000 r04=02100ee0 r05=02101388 r06=021087a0 r07=02101388
[...]
Special Purpose Registers:
 lr=0117f6e8 cr=04002022 xer=00000000 ctr=00000000
[...]
Floating Point Registers
 f0=00000000 00000000  00000000 00000000  f1=00000000 00000000  00000000 00000000
[...]
Memory:
 Stack top       : 0x00000000
[...]
+++STACK
0x011c1434
---STACK
---ID
+++ID TGID 100, Core 0, Thread 5 State 00000000, Sched: 48000000 Running
[...]
```

Decoding Lightweight Core Files

- BG/P: bgp_stack [exename] [corefile]
- VEAS: /soft/debuggers/scripts/bin/bgq_stack [exename] [corefile]

```
-----------------------------------------------------------------------
+++ID TGID 100, Core 0, Thread 5 State 00000000, Sched: 48000000 Running

0x01001534
foo
/gpfs/home/rloy/public/winter-workshop-2012/hellompi.c:33

0x01001704
main
/gpfs/home/rloy/public/winter-workshop-2012/hellompi.c:88

0x01194a44
generic_start_main
../csu/libc-start.c:231

0x01194cb8
__libc_start_main
../sysdeps/unix/sysv/linux/powerpc/libc-start.c:137

0xfffffffc
??
??:0
```

Decoding Lightweight Core Files (2)

- What's this other stuff? (MPI threads)

```
-----------------------------------------------------------------------
+++ID TGID 100, Core 0, Thread 1 State 40000000, Sched: 48000000

0x011c1434
clone
??:0

-----------------------------------------------------------------------
+++ID TGID 100, Core 1, Thread 2 State 40000000, Sched: 08000000

0x011c1434
clone
??:0

-----------------------------------------------------------------------
+++ID TGID 100, Core 2, Thread 3 State 40000000, Sched: 08000000

0x011c1434
clone
??:0

-----------------------------------------------------------------------
+++ID TGID 100, Core 3, Thread 4 State 40000000, Sched: 08000000

0x011c1434
clone
??:0
```

Outline

- bgp_stack
- coreprocessor
- gdb
- Allinea DDT
- TotalView

coreprocessor

- Useful when you have a large set of core files
  - Shows symbolic backtrace
  - Groups ranks that aborted in the same location together
  - *Can also attach to a running job to take snapshot*

- Location
  - BG/P: /bgsys/drivers/ppcfloor/tools/coreprocessor/coreprocessor.pl
    - Or link also at: /soft/apps/coreprocessor.pl
    - Attaching to running job requires administrator
  - VEAS: /bgsys/drivers/ppcfloor/coreprocessor
    - Attaching to running job should not require administrator (untested)

- Instructions:
  - www.alcf.anl.gov/resource-guides/debugging ("coreprocessor")

coreprocessor window

File Control Analyze Filter Sessions

Group Mode: Stack Traceback (condensed) Session 1 (MMC

```
0 :Compute Node (128)
1 :     0xfffffffc (128)
2 :         __libc_start_main (32)
3 :             generic_start_main (32)
4 :                 main (16)
5 :                     Allgather (16)
6 :                         PMPI_Allgather (16)
7 :                             MPIDO_Allgather (8)
8 :                                 MPIDO_Allreduce (8)
9 :                                     MPID_Progress_wait (1)
10:                                         DCMF_CriticalSection_cycle (1)
9 :                                     MPID_Progress_wait (7)
10:                                         DCMF_Messager_advance (1)
11:                                             DCMF::Queueing::Lockbox::Device::advance() (1)
10:                                         DCMF_Messager_advance (1)
11:                                             DCMF::Queueing::Tree::Device::advance() (1)
10:                                         DCMF_Messager_advance (5)
11:                                             DCMF::DMA::Device::advance() (2)
12:                                                 DCMF::DMA::RecFifoGroup::advance() (2)
13:                                                     DMA_RecFifoSimplePollNormalFifoById (2)
11:                                             DCMF::DMA::Device::advance() (3)
7 :                             MPIDO_Allgather (8)
8 :                                 MPIDO_Allreduce (8)
9 :                                     MPIR_Allreduce (8)
10:                                         MPIC_Sendrecv (8)
11:                                             MPID_Progress_wait (8)
12:                                                 DCMF_Messager_advance (8)
13:                                                     DCMF::Queueing::GI::Device::advance() (1)
13:                                                     DCMF::DMA::Device::advance() (3)
14:                                                         DCMF::DMA::RecFifoGroup::advance() (3)
15:                                                             DMA_RecFifoSimplePollNormalFifoById (3)
```

Outline

- bgp_stack
- coreprocessor
- gdb
- Allinea DDT
- Totalview

gdb

- Can connect single gdb client to single rank of your job
- BG/P
  - Connect as many as you like, but each is completely independent
  - http://www.alcf.anl.gov/resource-guides/debugging
- BG/Q
  - Will be supported
  - Limitations of CDTI (Common Debug and Tool Interface)
    - Each instance of gdb client counts as a “debug tool”
    - Only 4 tools may be connected to a job
      - Therefore at most 4 ranks can be examined
  - *Generally DDT or TotalView will be more useful*

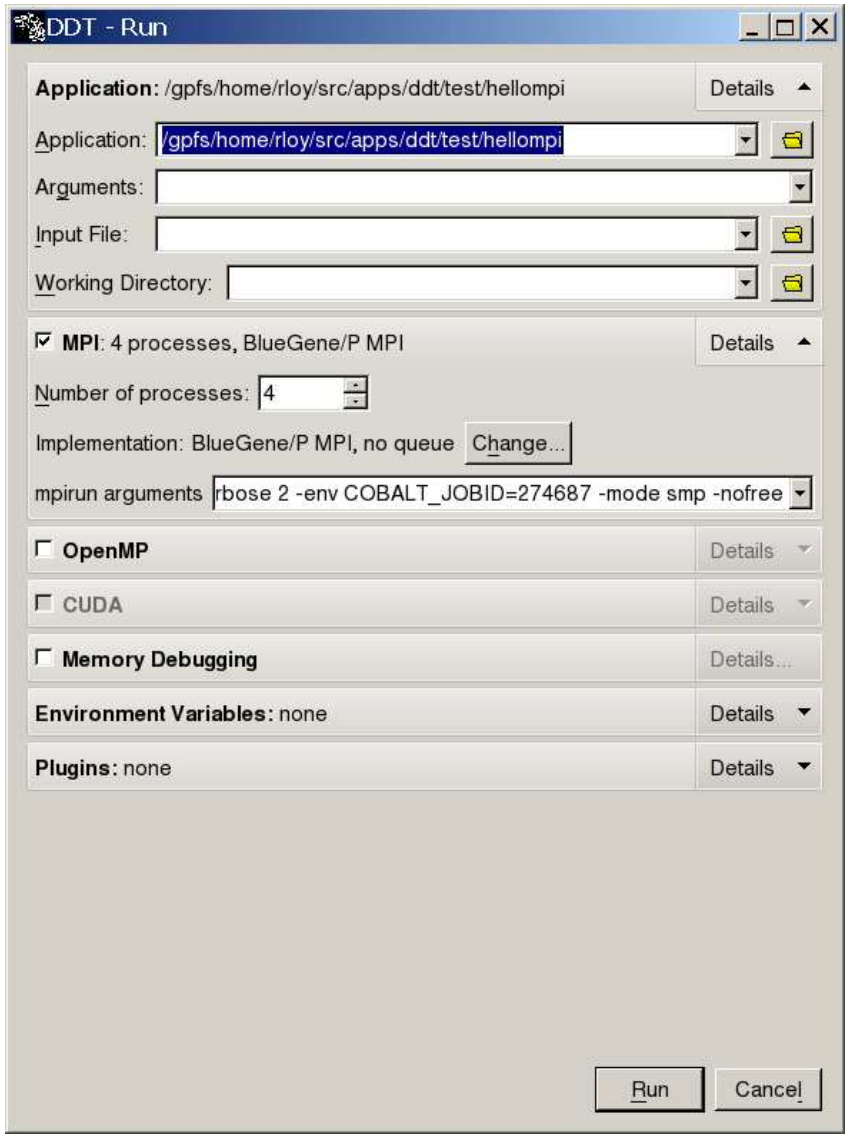
Outline

- bgp_stack
- coreprocessor
- gdb
- Allinea DDT
- Totalview

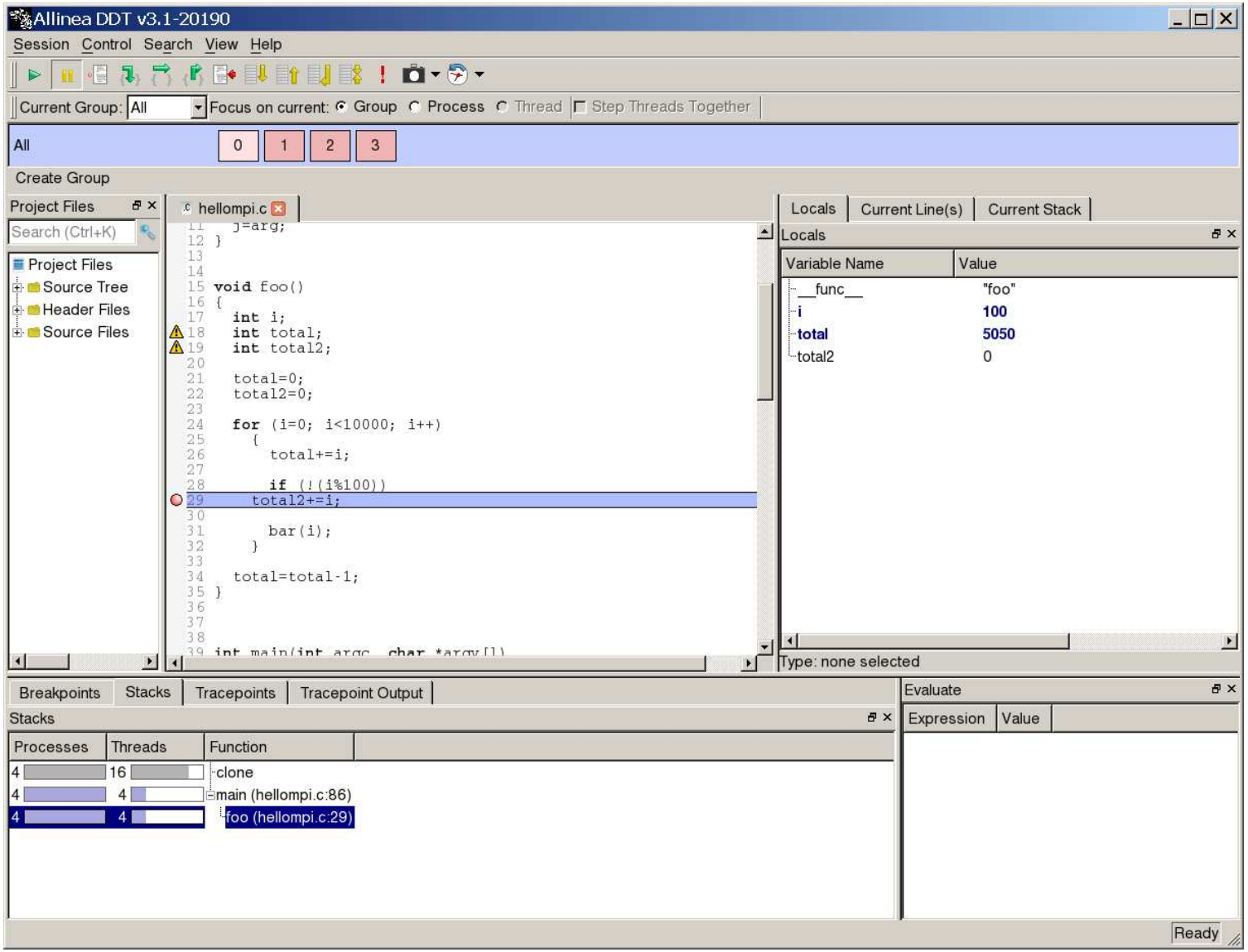
Allinea DDT

- BG/P licensing
  - 64-process permanent license
  - Full machine development license available (contact support)
- Startup overview
  - Compile –g –O0
    - OMP code compile -qsmp=omp:noauto:noopt
  - Softenv key "+ddt"
  - Need X11 server and ssh –X forwarding
  - Start interactive job with *isub*
  - Run ddt from isub
- More details:
  - http://www.alcf.anl.gov/resource-guides/debugging ("Using Allinea DDT")

Allinea DDT – Initial Windows



Allinea DDT – Main Window



Allinea DDT: The Path to BG/Q

- 2010: Prior work
  - Improved scaling through tree communication structure between front-end application and DDT servers on IONs . Limited to 32 processes per ION.
- 2010-11: Scalability Phase 1 (BG/P)
  - Reduce memory use on single ION by sharing a single copy of program symbol table amongst all ranks. Up to 128 processes per ION.
  - Note: one DDT server process and one gdbserver process per user process
- 2011-12: Scalability Phase 2 (BG/P)
  - Multiplex DDT server processes
  - Multiplex gdbserver processes
  - Up to 256 processes per ION. Development version now available
- Port to BG/Q
  - *In progress*

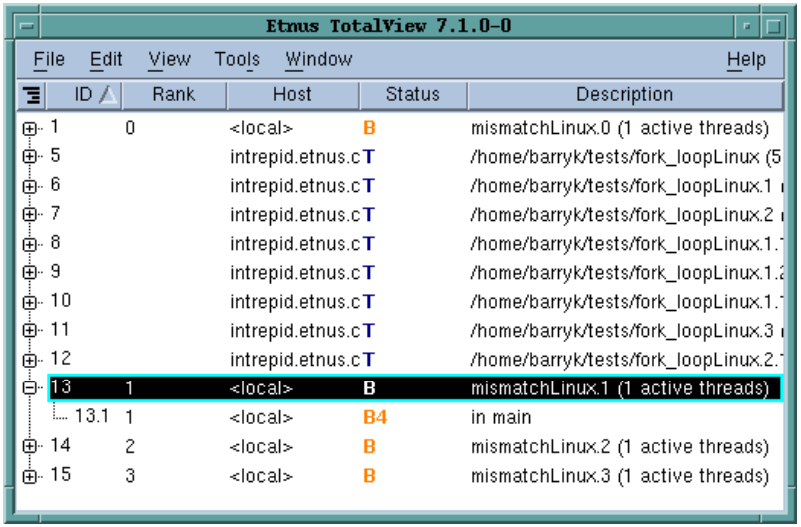
Outline

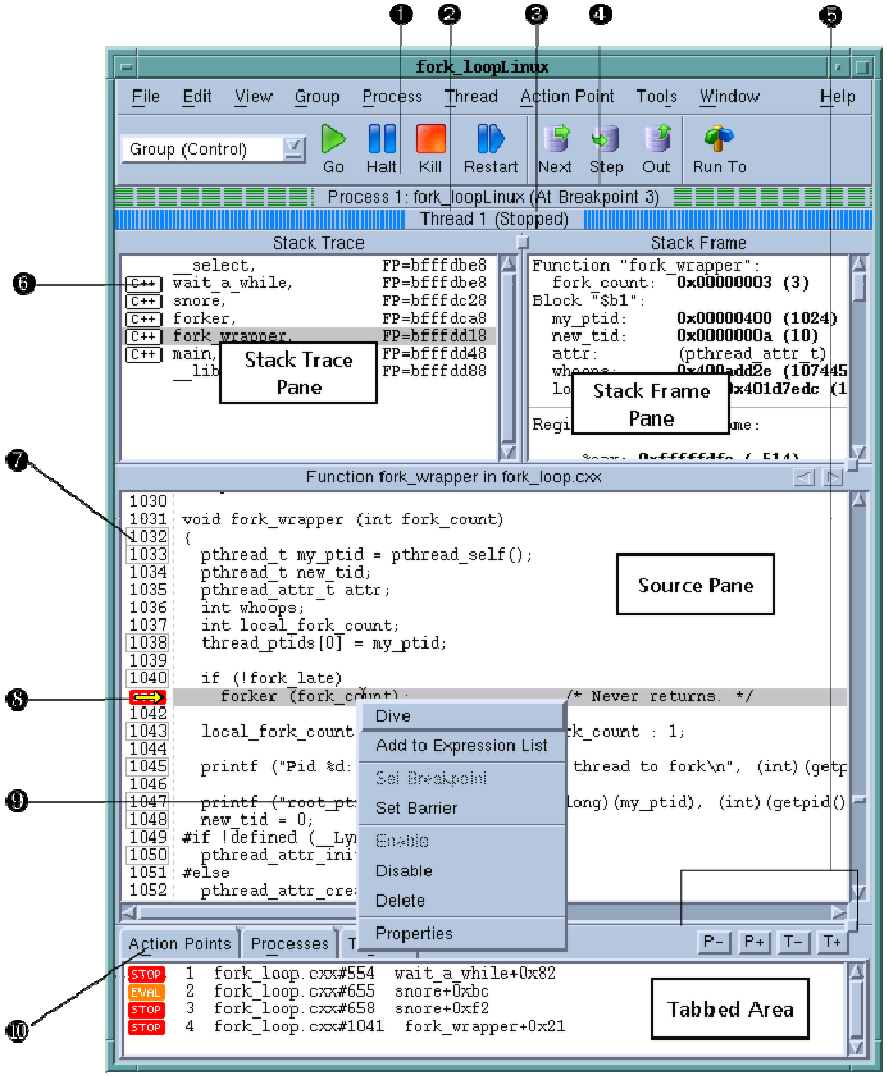
- bgp_stack
- coreprocessor
- gdb
- Allinea DDT
- Totalview

TotalView

- BG/P License
  - 2048 processes
  - Latest version available 8.9.0.0
- Startup overview
  - Compile –g –O0
    - OMP code compile -qsmp=omp:noauto:noopt
  - Softenv key “+totalview”
  - Need X11 server and ssh –X forwarding
  - Start interactive job with *isub*
  - Run totalview from isub
- More details:
  - http://www.alcf.anl.gov/resource-guides/debugging (“Using TotalView”)

TotalView





1. Process ID (PID)
2. Thread ID (TID)
3. Thread status
4. Process status
5. Process/thread switching controls
6. Language of routine
7. Line number area
8. Current program counter
9. Context menu
10. Action Points tab displayed

TotalView on BG/Q

- June 2011 - Preliminary work with CDTI
- Aug 2011 - Initial porting on BG/Q
- Oct 2011 - Basic functions working
  - Includes breakpoints, watchpoints, stacks, single stepping, QPX, static/dynamic and shared libraries, job control, fast conditional breakpoints
  - Some advanced functionality still in progress:
    - scalability testing, fast conditional watchpoints, TM/SE regions
- Research license for VEAS – in progress, expected soon