

# BG/Q Parallel Debugging *Ready to Scale*

ALCF MiraCon  
March 4, 2013

Ray Loy  
Applications Performance Engineering  
ALCF

# Outline

- `bgq_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
  - <https://www.alcf.anl.gov/resource-guides/vesta-debugging-core-files>



# 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

- `bgp_stack [exename] [corefile]`
- `bgq_stack [optional_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  
  
0xffffffffc  
??  
??: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

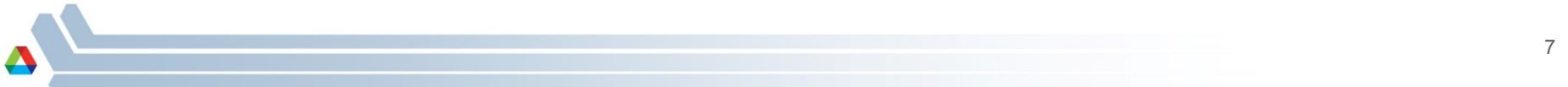
0x011c1434  
clone  
?:0

-----  
+++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

- bgq\_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: /soft/apps/coreprocessor.pl
    - Attaching to running job requires administrator
  - BG/Q: coreprocessor.pl is in your default PATH
    - Attaching to running job does **not** require administrator
    - `coreprocessor -nogui -snapshot=<filename> -j=<jobid>`
- Scalability limit
  - Maximum 32K ranks (*LLNL STAT coming soon*)
- Instructions:
  - BG/P [www.alcf.anl.gov/resource-guides/coreprocessor](http://www.alcf.anl.gov/resource-guides/coreprocessor)
  - BG/Q Application Developer Redbook (draft)
    - <http://www.redbooks.ibm.com/redpieces/abstracts/sg247948.html?Open>



# coreprocessor window

```
File Control Analyze Filter Sessions
Group Mode: Stack Traceback (condensed) Session 1 (MMC)
0 : Compute Node (128)
1 :   0xffffffffc (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_Messenger_advance (1)
11:                       DCMF::Queueing::Lockbox::Device::advance() (1)
10:                     DCMF_Messenger_advance (1)
11:                       DCMF::Queueing::Tree::Device::advance() (1)
10:                     DCMF_Messenger_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_Messenger_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

- bgq\_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-and-profiling>
- BG/Q
  - 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
- Start a debug session using *isub*
  - `isub -q default -t 30 -n 64`
- gdb can also load a compute-node **binary** corefile
- Generally DDT or TotalView will be more useful



# Outline

- bgq\_stack
- coreprocessor
- gdb
- Allinea DDT
- TotalView



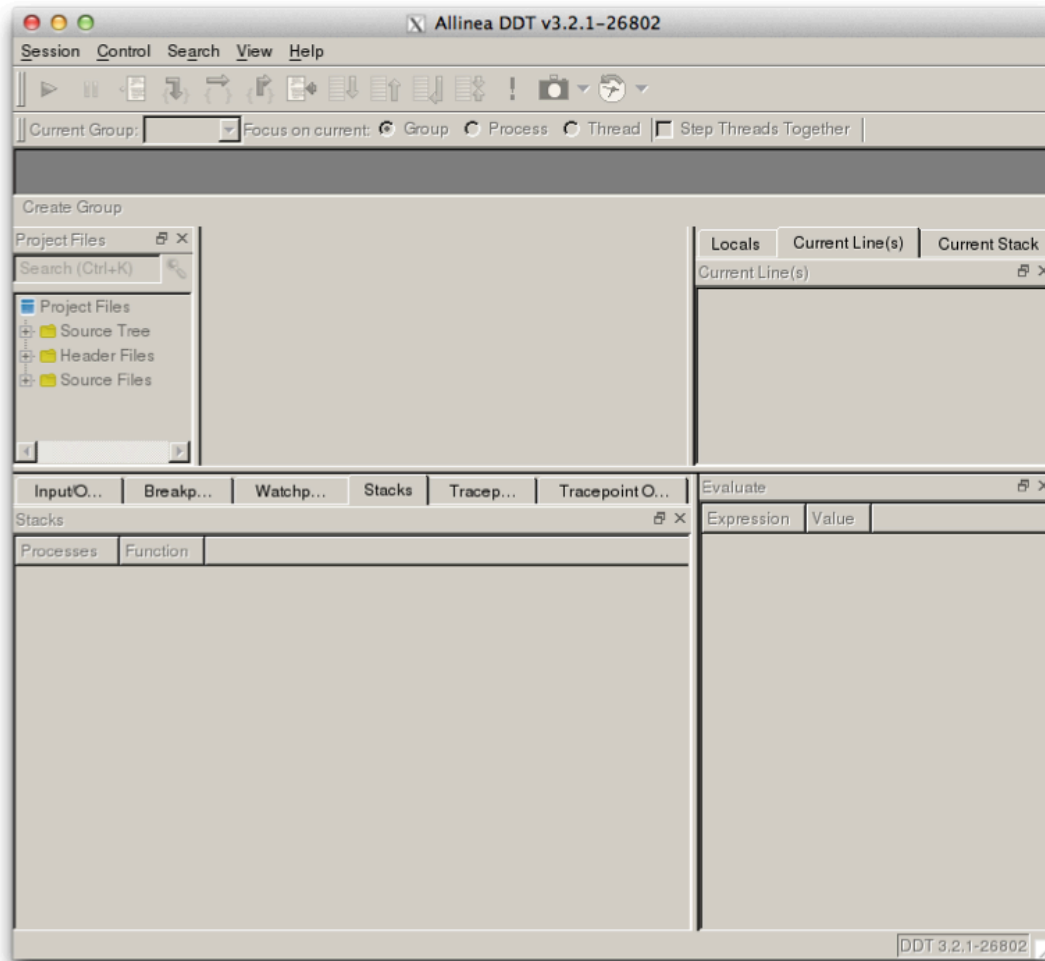
# Allinea DDT

- Licensing
  - 32K-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
  - [BG/P only] Start interactive job with *isub*
  - [BG/P or BG/Q] Run ddt and submit job through GUI
- More details:
  - [BG/P] <http://www.alcf.anl.gov/resource-guides/allinea-ddt>



# Allinea DDT

## Main window



## Welcome window



Click here

Screen shots courtesy M. Garcia



Customize the windows with your data

DDT - Run

**Application:** /home/mgarcia/TEST/executable.out Details...

Application: /home/mgarcia/TEST/executable.out Folder icon

Arguments: Dropdown arrow

Input File: Folder icon

Working Directory: /home/mgarcia/TEST Folder icon

**MPI:** 1 node, BlueGene/Q Details...

Number of nodes (16 processes per node)  Dropdown arrow

Implementation: BlueGene/Q, no queue Change... Click here

runjob arguments Dropdown arrow

**OpenMP** Details...

**CUDA** Details...

**Memory Debugging** Details...

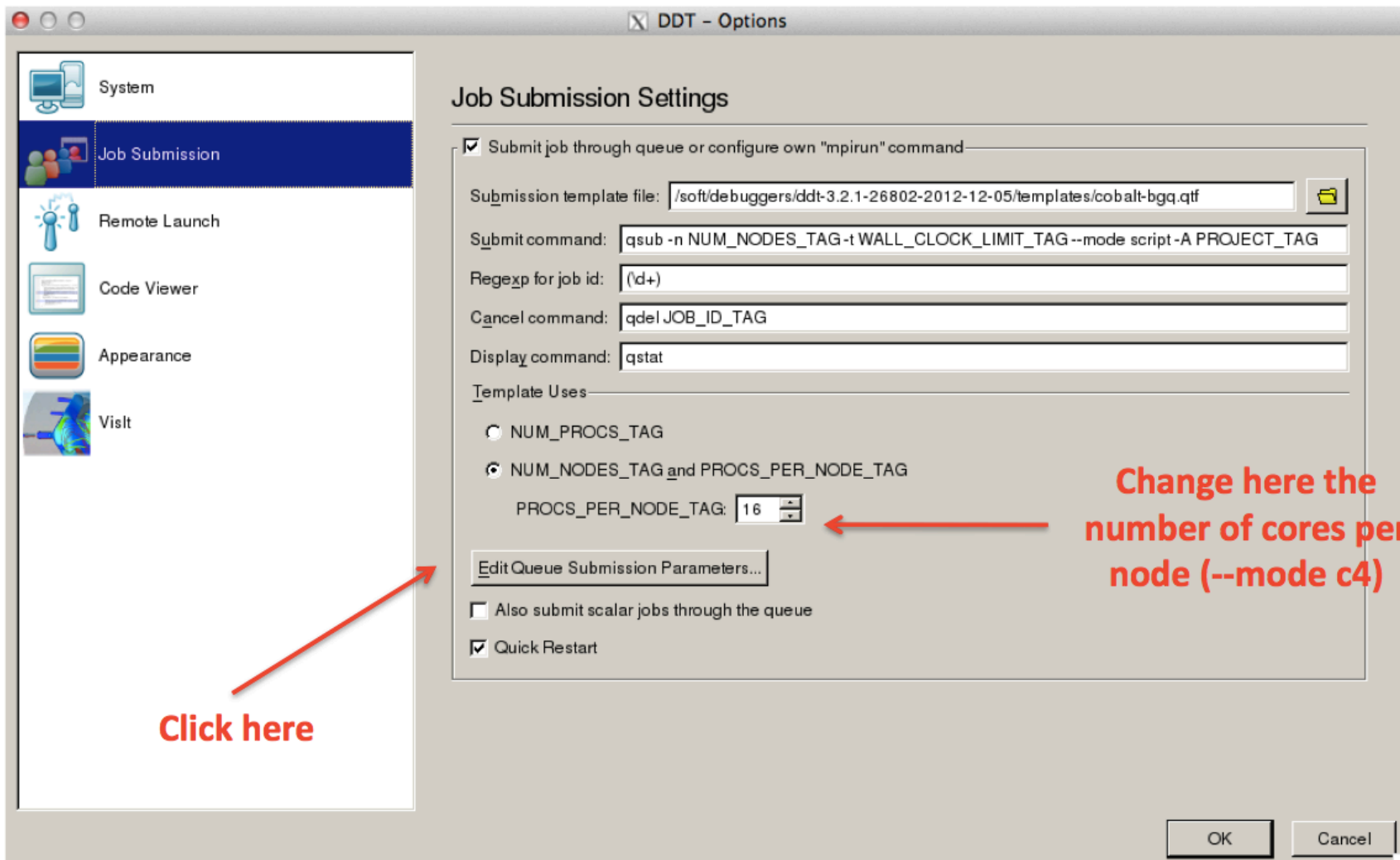
**Environment Variables:** none Details...

**Plugins:** none Details...

Run Cancel







Click here

Change here the number of cores per node (--mode c4)

Queue Submission Parameters

Project: Catalyst

Wall Clock Limit (minutes): 30

OK Cancel

**Customize  
your info**

- Click OK for the two popup windows, then "Run"



# Outline

- bgq\_stack
- coreprocessor
- gdb
- Allinea DDT
- **TotalView**



# TotalView

- Licensing
  - BG/P: 2048 processes (Latest version available 8.9.0.0)
  - BG/Q: 8192 processes (demo license expires 12-Apr-2013)
- Startup overview
  - Compile `-g -O0`
    - OMP code compile `-qsmp=omp:noauto:noopt`
  - BG/P: softenv key “+totalview” or BG/Q: `/soft/debuggers/totalview`
  - Need X11 server and ssh `-X` forwarding
  - [BG/P] Start interactive job with `isub`
  - [BG/Q] Copy job scripts from `/soft/debuggers/scripts/totalview-examples`
- More details:
  - [BG/P] <http://www.alcf.anl.gov/resource-guides/totalview>



# TotalView Scripts `/soft/debuggers/scripts/totalview-examples/`

- To submit:

```
#!/bin/bash
qsub -t 60 -n 128 --mode script -O LOG --env DISPLAY=$DISPLAY ./runtv.sh
echo "After your job starts, do a 'tail -f LOG' to see output"
```

- The job script `runtv.sh` :

```
#!/bin/sh
# Modify the totalview arguments for your situation
```

```
echo "Starting Cobalt job script"
echo "DISPLAY is $DISPLAY"
```

```
/soft/debuggers/totalview -args runjob -p 1 -n 128 --block
$COBALT_PARTNAME --verbose 2 --envs
PAMID_VERBOSE=1 : yourprogram.exe
```

