ARM Forge DDT

Quick intro to Debugging with Forge DDT

Presented by Will Castillo

Intro

- ARM Forge DDT (Distributed Debugging Tool)
- Commercial debugging tool originally developed by Allinea Software company, 2002
- Graphical interface to debug serial or highly parallelized codes within HPC
- As of 2016 DDT was used on 20 of the 25 fastest supercomputers in the world

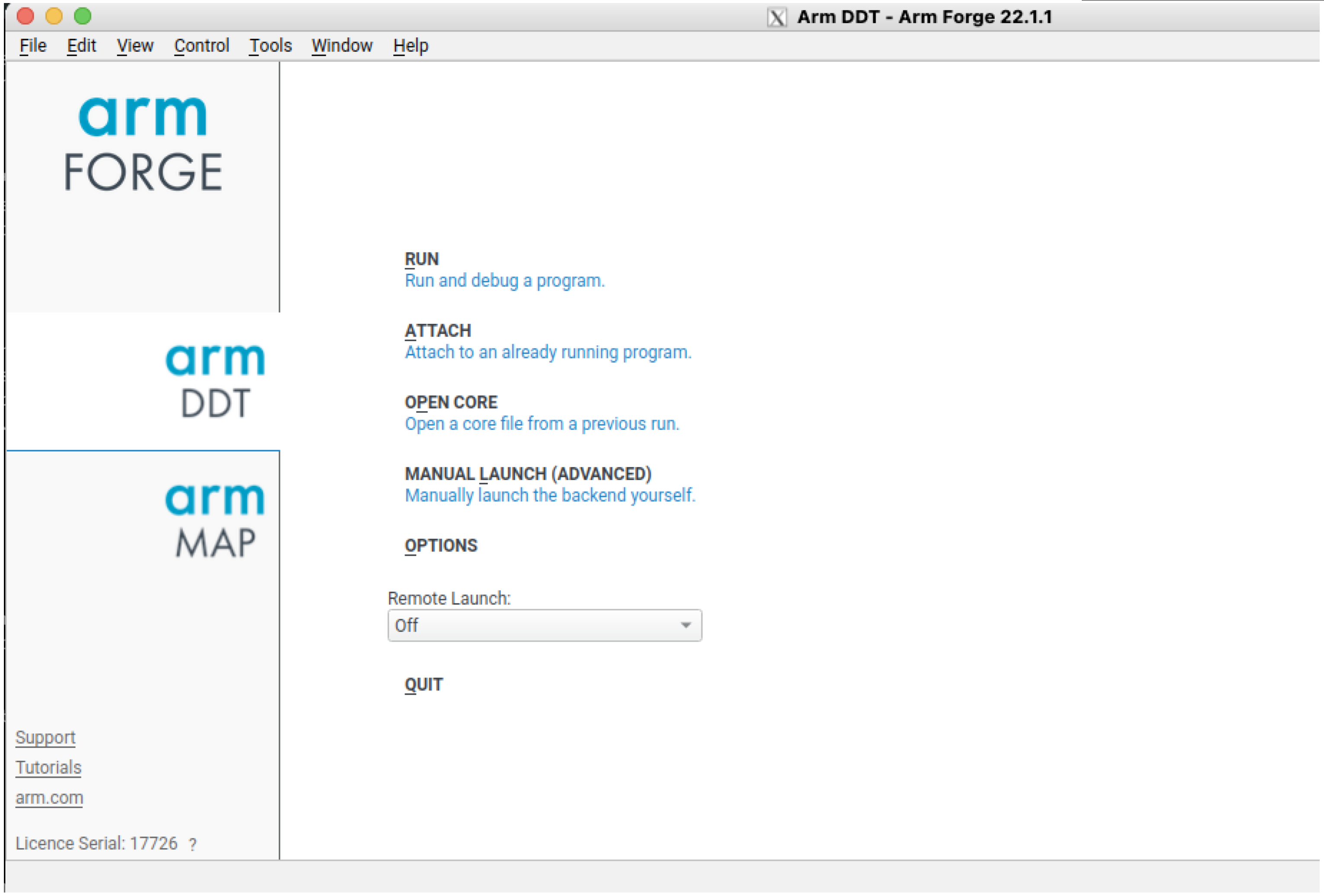
Paradigms

- Supports single and multithreaded processes
- OpenMP
- MPI
- Heterogenous software (GPU software)
- Hybrid codes e.g. MPI with OpenMP *or* MPI with CUDA

Language Support

- C
- C++
- All flavors of Fortran, including f90
- Python (limited)
- GPU languages (CUDA, hipcc)

A look into…



Arm DDT - Arm Forge 22.1.1

RUN
Run and debug a program.

ATTACH
Attach to an already running program.

OPEN CORE
Open a core file from a previous run.

MANUAL LAUNCH (ADVANCED)
Manually launch the backend yourself.

OPTIONS

Remote Launch: Off

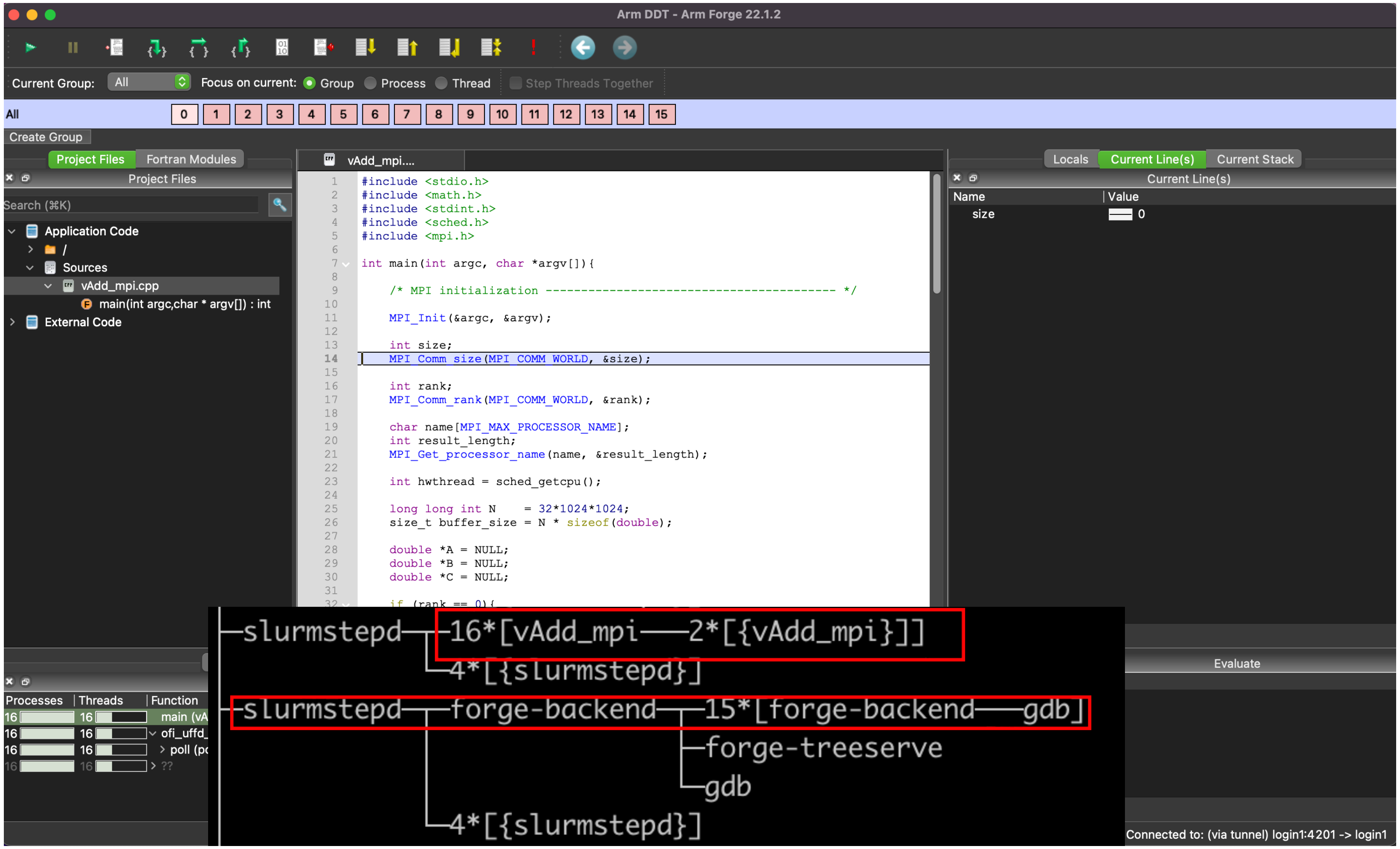
QUIT

Support
Tutorials
arm.com

Licence Serial: 17726 ?

Connecting

- Backend connects to all ranks



```
#include <stdio.h>
#include <math.h>
#include <stdint.h>
#include <sched.h>
#include <mpi.h>

int main(int argc, char *argv[]){

    /* MPI initialization ----------------------------------------- */

    MPI_Init(&argc, &argv);

    int size;
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    int rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    char name[MPI_MAX_PROCESSOR_NAME];
    int result_length;
    MPI_Get_processor_name(name, &result_length);

    int hwthread = sched_getcpu();

    long long int N     = 32*1024*1024;
    size_t buffer_size = N * sizeof(double);

    double *A = NULL;
    double *B = NULL;
    double *C = NULL;

    if (rank == 0){
```

Name	Value
size	0

Processes	Threads	Function
16	16	main (vA
16	16	ofi_uffd_
16	16	poll (po
16	16	??

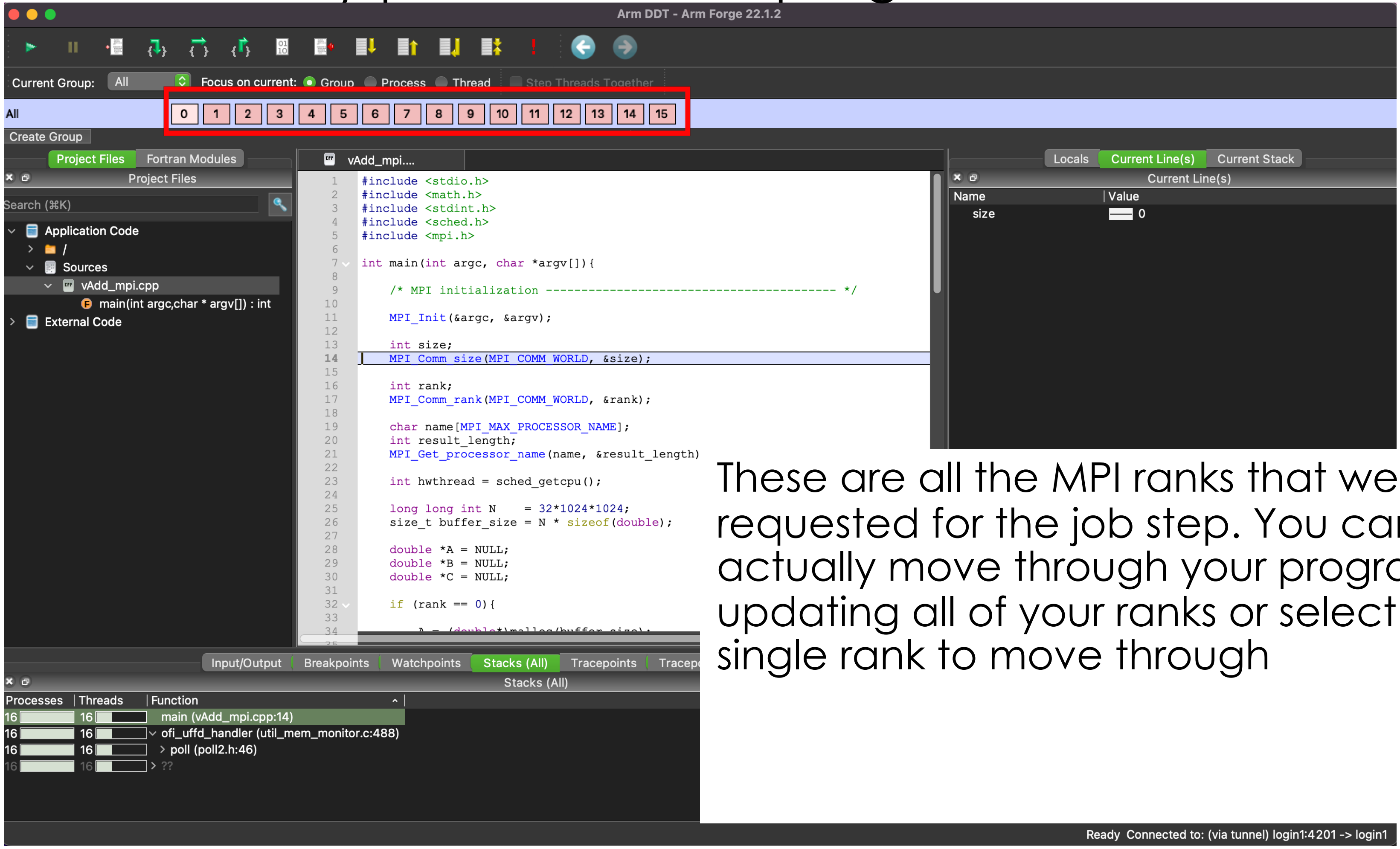
Connected to: (via tunnel) login1:4201 -> login1

```
├─slurmstepd─┬─16*[vAdd_mpi───2*[{vAdd_mpi}]]
│            └─4*[{slurmstepd}]
├─slurmstepd─┬─forge-backend─┬─15*[forge-backend───gdb]
│            │               ├─forge-treeserve
│            │               └─gdb
│            └─4*[{slurmstepd}]
```

Open slide master to edit

Functionality

- Control many processes of a program



All 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

```
#include <stdio.h>
#include <math.h>
#include <stdint.h>
#include <sched.h>
#include <mpi.h>

int main(int argc, char *argv[]){

    /* MPI initialization ----------------------------------------- */

    MPI_Init(&argc, &argv);

    int size;
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    int rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    char name[MPI_MAX_PROCESSOR_NAME];
    int result_length;
    MPI_Get_processor_name(name, &result_length);

    int hwthread = sched_getcpu();

    long long int N     = 32*1024*1024;
    size_t buffer_size = N * sizeof(double);

    double *A = NULL;
    double *B = NULL;
    double *C = NULL;

    if (rank == 0){
```

Current Line(s)

Name	Value
size	0

Stacks (All)

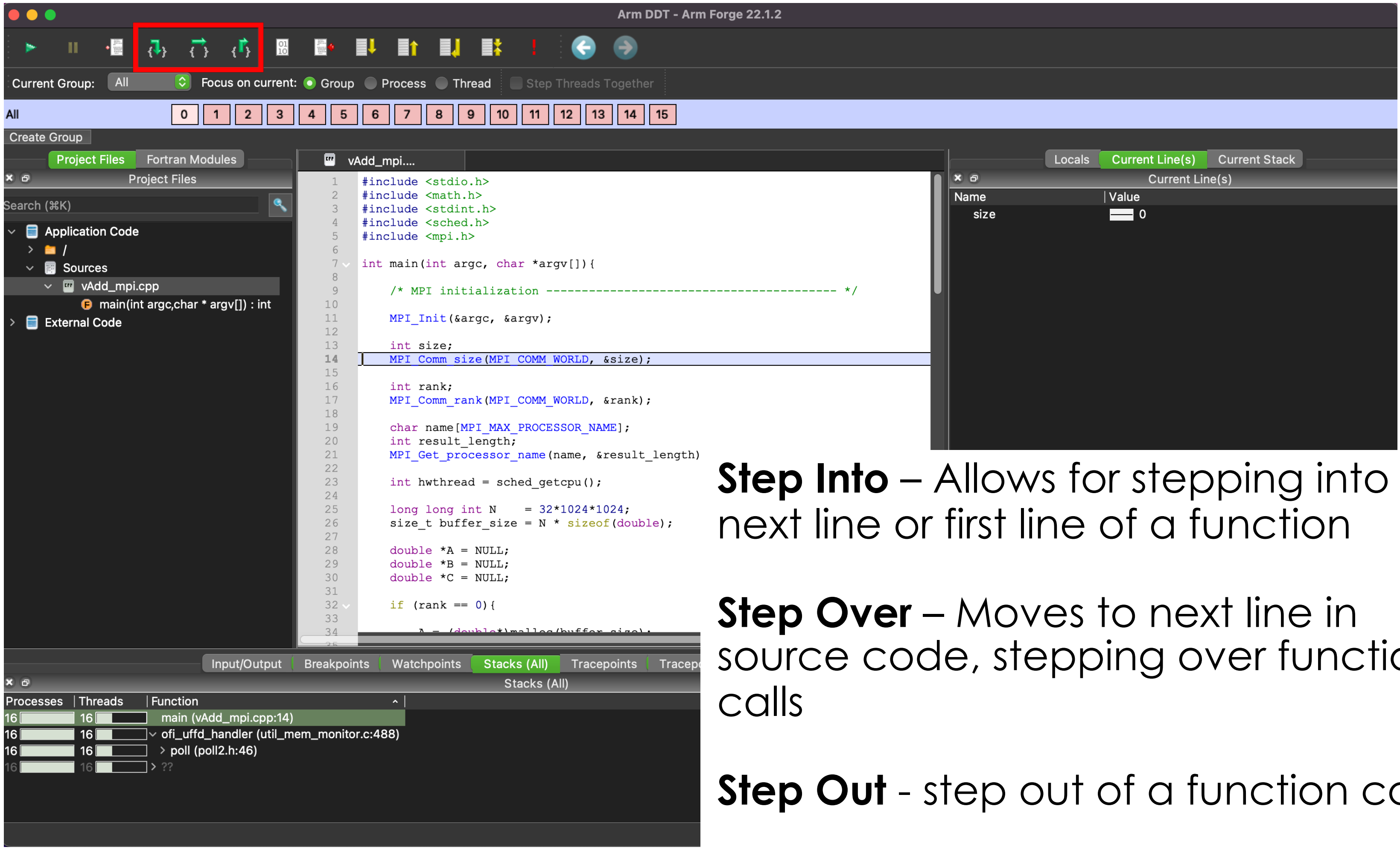
Processes	Threads	Function
16	16	main (vAdd_mpi.cpp:14)
16	16	ofi_uffd_handler (util_mem_monitor.c:488)
16	16	poll (poll2.h:46)
16	16	??

Ready Connected to: (via tunnel) login1:4201 -> login1

These are all the MPI ranks that were requested for the job step. You can actually move through your program updating all of your ranks or select a single rank to move through

Functionality

- Allows the user to step through a program



```
#include <stdio.h>
#include <math.h>
#include <stdint.h>
#include <sched.h>
#include <mpi.h>

int main(int argc, char *argv[]){

    /* MPI initialization ---------------------------------------- */

    MPI_Init(&argc, &argv);

    int size;
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    int rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    char name[MPI_MAX_PROCESSOR_NAME];
    int result_length;
    MPI_Get_processor_name(name, &result_length)

    int hwthread = sched_getcpu();

    long long int N     = 32*1024*1024;
    size_t buffer_size = N * sizeof(double);

    double *A = NULL;
    double *B = NULL;
    double *C = NULL;

    if (rank == 0){
```

Current Line(s)

Name	Value
size	0

Stacks (All)

Processes	Threads	Function
16	16	main (vAdd_mpi.cpp:14)
16	16	ofi_uffd_handler (util_mem_monitor.c:488)
16	16	poll (poll2.h:46)
16	16	??

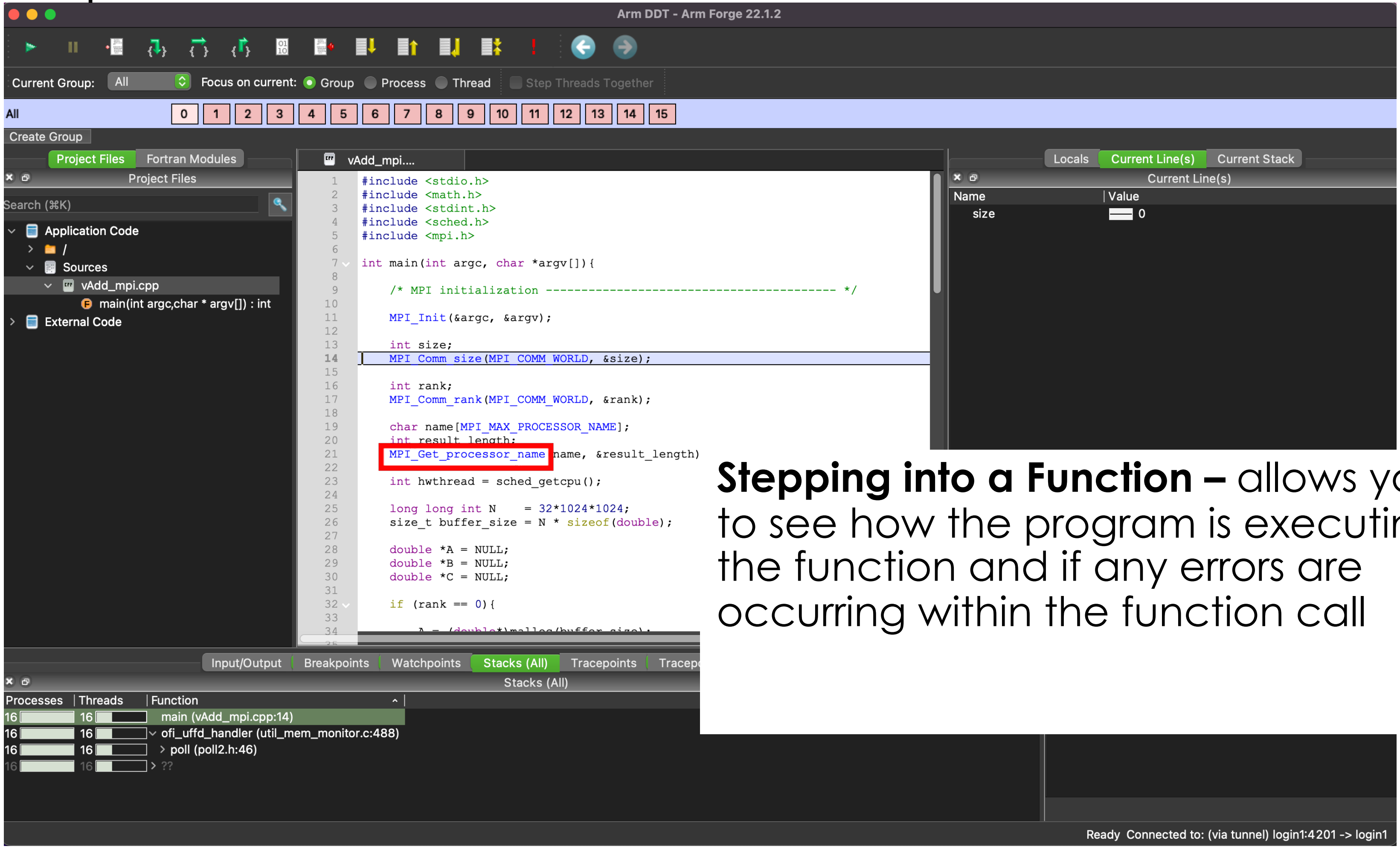
Step Into – Allows for stepping into the next line or first line of a function

Step Over – Moves to next line in source code, stepping over function calls

Step Out - step out of a function call

Functionality

- Step into Functions



```
#include <stdio.h>
#include <math.h>
#include <stdint.h>
#include <sched.h>
#include <mpi.h>

int main(int argc, char *argv[]){

    /* MPI initialization ----------------------------------------- */

    MPI_Init(&argc, &argv);

    int size;
    MPI_Comm_size(MPI_COMM_WORLD, &size);

    int rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    char name[MPI_MAX_PROCESSOR_NAME];
    int result_length;
    MPI_Get_processor_name(name, &result_length)

    int hwthread = sched_getcpu();

    long long int N     = 32*1024*1024;
    size_t buffer_size = N * sizeof(double);

    double *A = NULL;
    double *B = NULL;
    double *C = NULL;

    if (rank == 0){
```

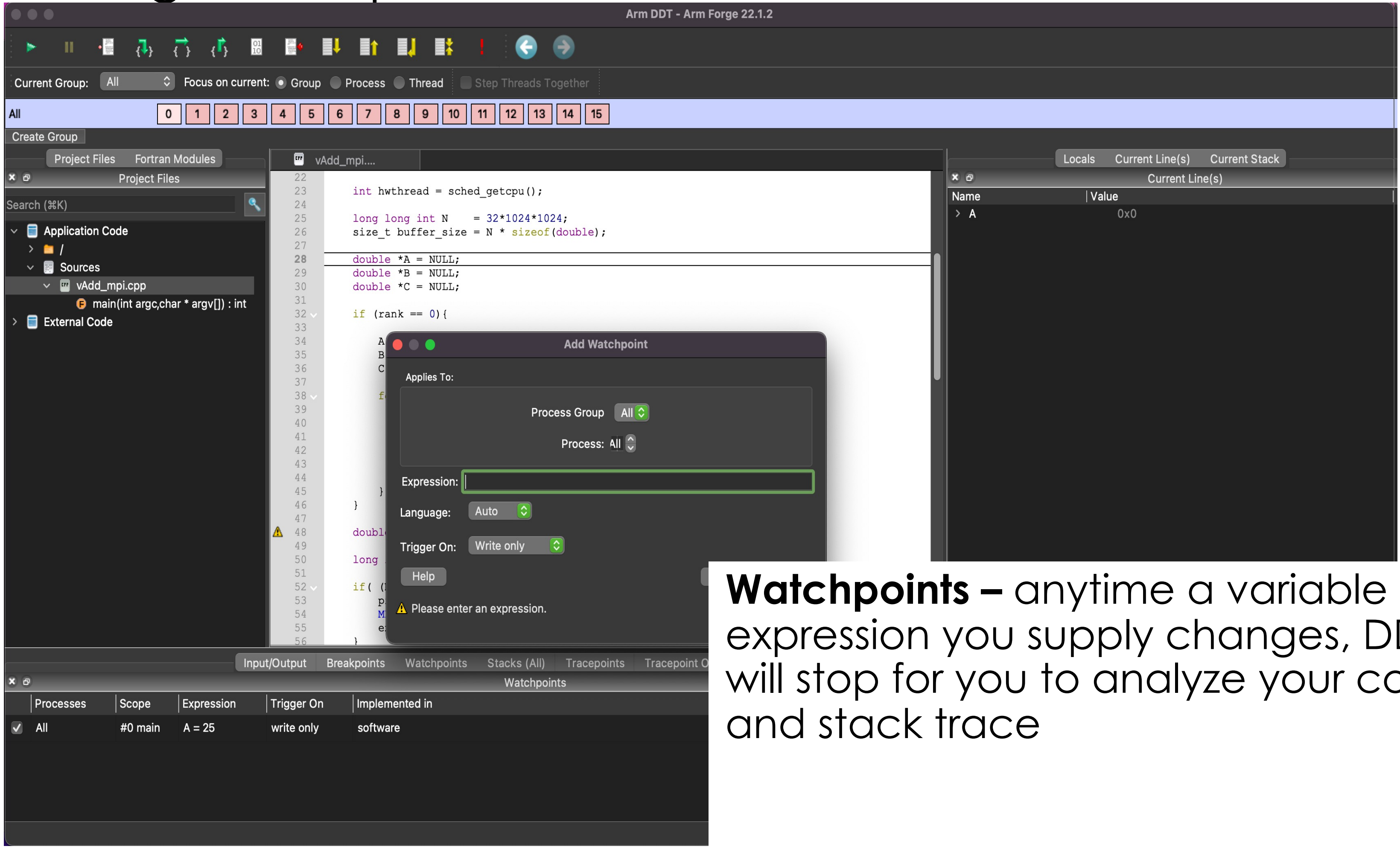
Name	Value
size	0

Processes	Threads	Function
16	16	main (vAdd_mpi.cpp:14)
16	16	ofi_uffd_handler (util_mem_monitor.c:488)
16	16	poll (poll2.h:46)
16	16	??

Stepping into a Function – allows you to see how the program is executing the function and if any errors are occurring within the function call

Functionality

- Setting watchpoints



Arm DDT - Arm Forge 22.1.2

Current Group: All   Focus on current: ● Group ○ Process ○ Thread   Step Threads Together

All 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Create Group

Project Files
- Application Code
  - /
  - Sources
    - vAdd_mpi.cpp
      - main(int argc,char * argv[]) : int
- External Code

vAdd_mpi....

```
    int hwthread = sched_getcpu();

    long long int N    = 32*1024*1024;
    size_t buffer_size = N * sizeof(double);

    double *A = NULL;
    double *B = NULL;
    double *C = NULL;

    if (rank == 0){

        A
        B
        C

        f




        }
    }

    doubl

    long

    if( (
        p
        M
        e
    }
```

Add Watchpoint

Applies To:
Process Group: All
Process: All
Expression:
Language: Auto
Trigger On: Write only
Help
⚠ Please enter an expression.

Current Line(s)

Name	Value
A	0x0

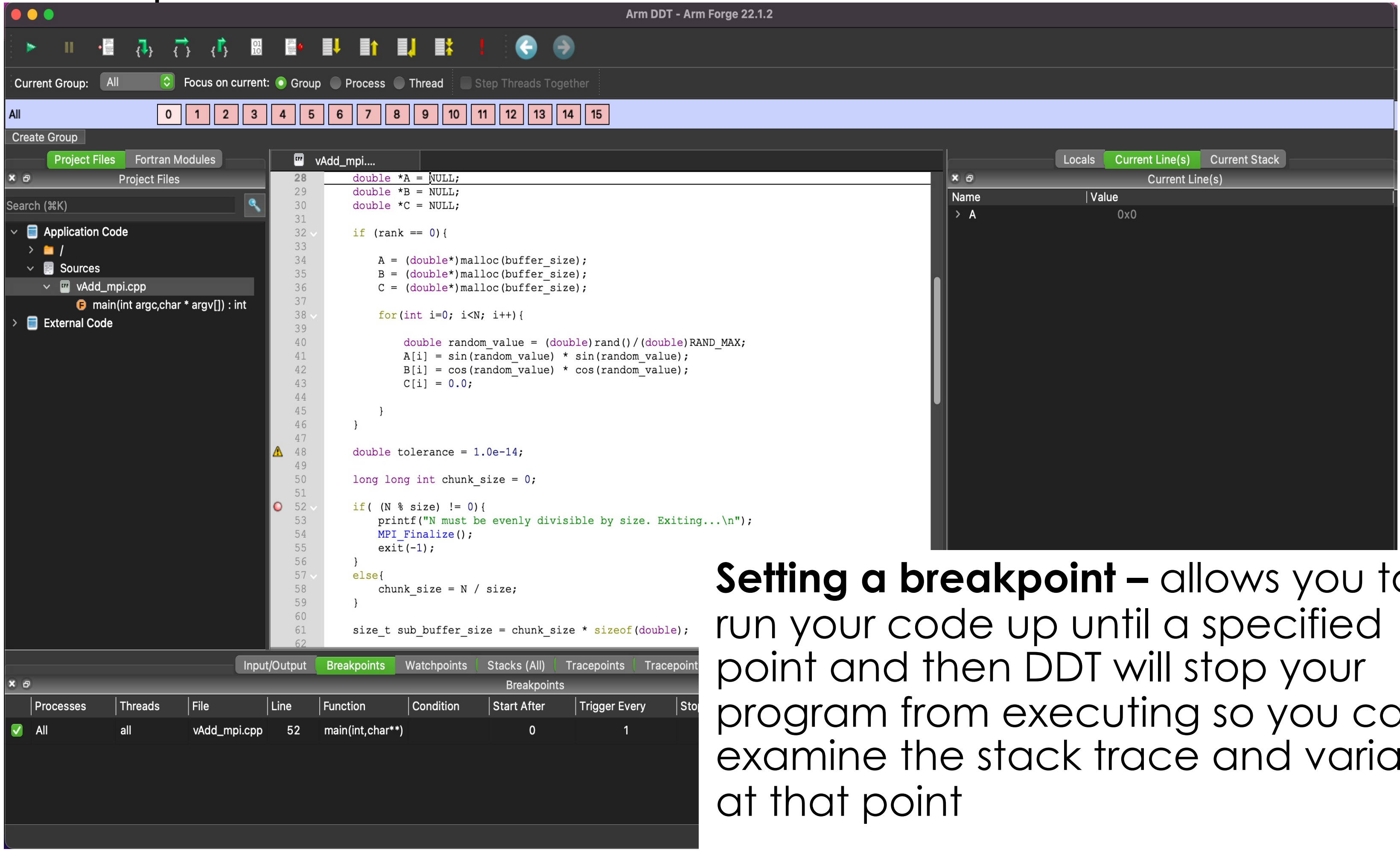
Watchpoints

	Processes	Scope	Expression	Trigger On	Implemented in
✓	All	#0 main	A = 25	write only	software

Watchpoints – anytime a variable or expression you supply changes, DDT will stop for you to analyze your code and stack trace

Functionality

- Breakpoints



```
28      double *A = NULL;
29      double *B = NULL;
30      double *C = NULL;
31
32      if (rank == 0){
33
34          A = (double*)malloc(buffer_size);
35          B = (double*)malloc(buffer_size);
36          C = (double*)malloc(buffer_size);
37
38          for(int i=0; i<N; i++){
39
40              double random_value = (double)rand()/(double)RAND_MAX;
41              A[i] = sin(random_value) * sin(random_value);
42              B[i] = cos(random_value) * cos(random_value);
43              C[i] = 0.0;
44
45          }
46      }
47
48      double tolerance = 1.0e-14;
49
50      long long int chunk_size = 0;
51
52      if( (N % size) != 0){
53          printf("N must be evenly divisible by size. Exiting...\n");
54          MPI_Finalize();
55          exit(-1);
56      }
57      else{
58          chunk_size = N / size;
59      }
60
61      size_t sub_buffer_size = chunk_size * sizeof(double);
62
```

Current Line(s)

Name	Value
A	0x0

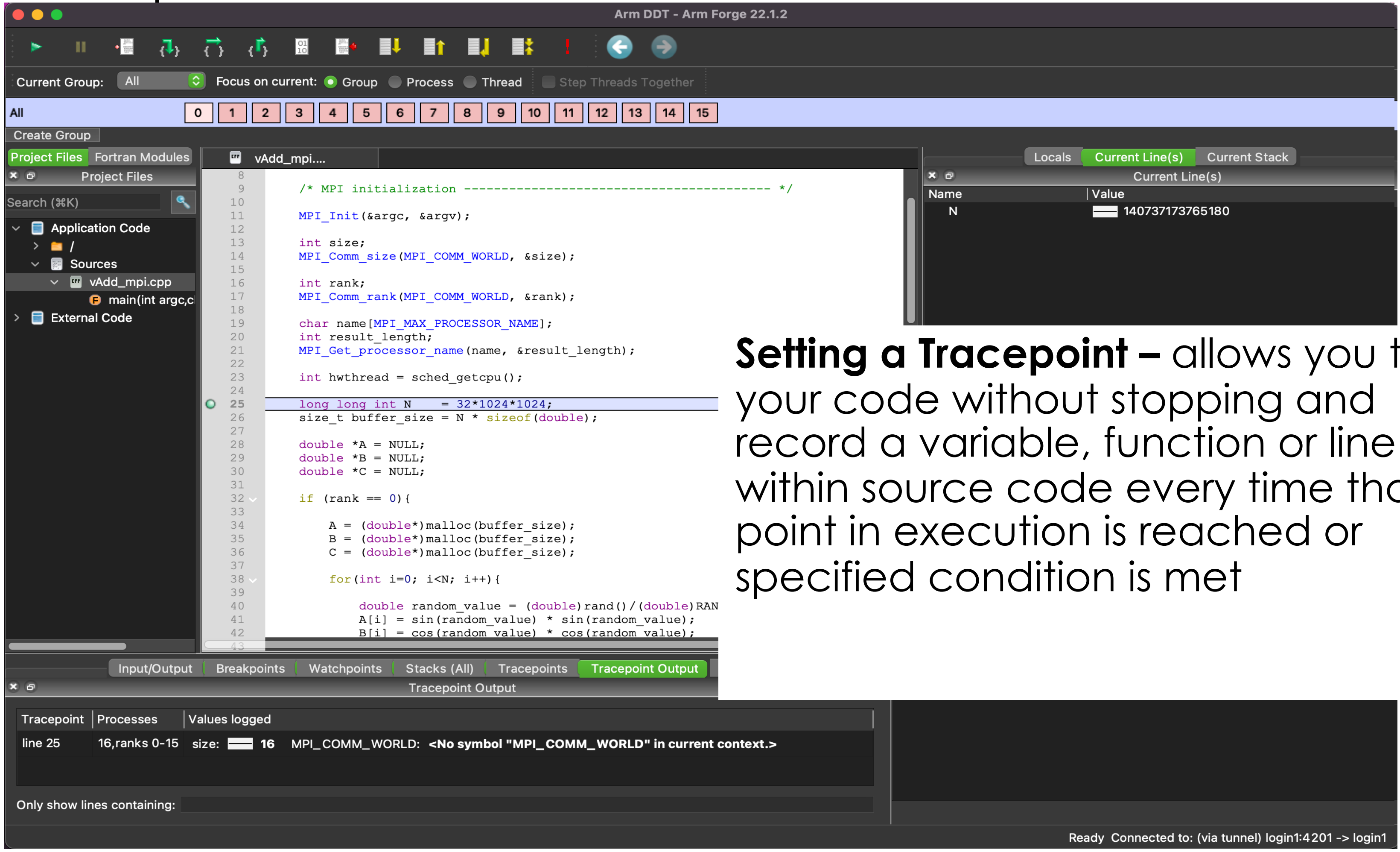
Breakpoints

	Processes	Threads	File	Line	Function	Condition	Start After	Trigger Every	Sto
✓	All	all	vAdd_mpi.cpp	52	main(int,char**)		0	1	

Setting a breakpoint – allows you to run your code up until a specified point and then DDT will stop your program from executing so you can examine the stack trace and variables at that point

Functionality

- Tracepoints



```
8
9      /* MPI initialization ----------------------------------------- */
10
11     MPI_Init(&argc, &argv);
12
13     int size;
14     MPI_Comm_size(MPI_COMM_WORLD, &size);
15
16     int rank;
17     MPI_Comm_rank(MPI_COMM_WORLD, &rank);
18
19     char name[MPI_MAX_PROCESSOR_NAME];
20     int result_length;
21     MPI_Get_processor_name(name, &result_length);
22
23     int hwthread = sched_getcpu();
24
25     long long int N     = 32*1024*1024;
26     size_t buffer_size = N * sizeof(double);
27
28     double *A = NULL;
29     double *B = NULL;
30     double *C = NULL;
31
32     if (rank == 0){
33
34         A = (double*)malloc(buffer_size);
35         B = (double*)malloc(buffer_size);
36         C = (double*)malloc(buffer_size);
37
38         for(int i=0; i<N; i++){
39
40             double random_value = (double)rand()/(double)RAN
41             A[i] = sin(random_value) * sin(random_value);
42             B[i] = cos(random_value) * cos(random_value);
```

Current Line(s)

Name	Value
N	140737173765180

Tracepoint Output

Tracepoint	Processes	Values logged
line 25	16,ranks 0-15	size: 16 MPI_COMM_WORLD: <No symbol "MPI_COMM_WORLD" in current context.>

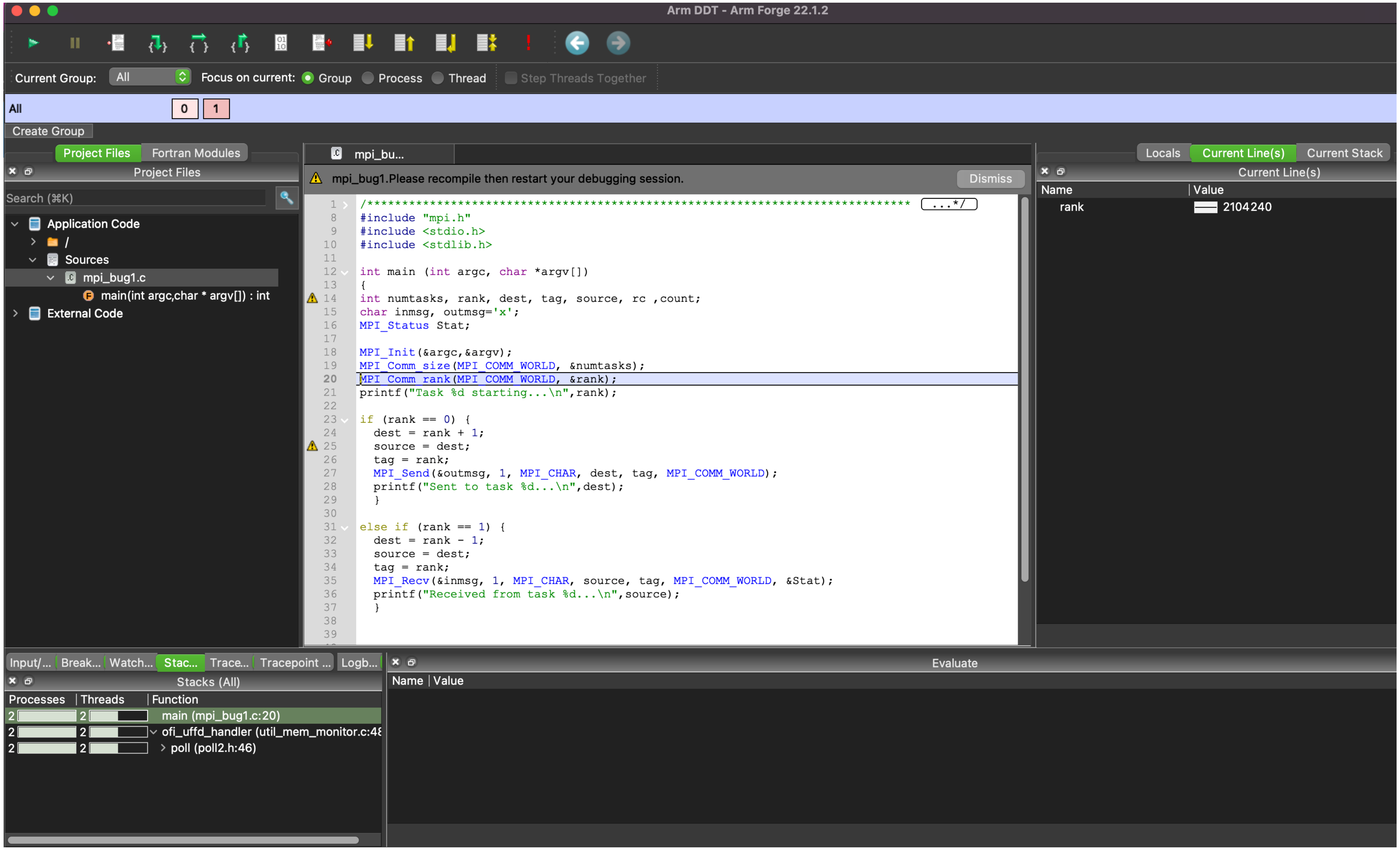
Setting a Tracepoint – allows you to run your code without stopping and record a variable, function or line within source code every time that point in execution is reached or specified condition is met

Conclusion

- Powerful debugger with graphical interface
- Multiple ways of connecting your program to DDT
- Start/stop features are critical when debugging codes at scale
- A competing tool is called TotalView

Demo

Initial connect



Arm DDT - Arm Forge 22.1.2

Current Group: All   Focus on current: (● Group) Process Thread   Step Threads Together

All   0   1

Create Group

Project Files | Fortran Modules

Project Files

- Application Code
  - /
  - Sources
    - mpi_bug1.c
      - main(int argc,char * argv[]) : int
- External Code

mpi_bu...

⚠ mpi_bug1.Please recompile then restart your debugging session.   Dismiss

```
/*********************************************************************** ...*/
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv[])
{
int numtasks, rank, dest, tag, source, rc ,count;
char inmsg, outmsg='x';
MPI_Status Stat;

MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
printf("Task %d starting...\n",rank);

if (rank == 0) {
  dest = rank + 1;
  source = dest;
  tag = rank;
  MPI_Send(&outmsg, 1, MPI_CHAR, dest, tag, MPI_COMM_WORLD);
  printf("Sent to task %d...\n",dest);
  }

else if (rank == 1) {
  dest = rank - 1;
  source = dest;
  tag = rank;
  MPI_Recv(&inmsg, 1, MPI_CHAR, source, tag, MPI_COMM_WORLD, &Stat);
  printf("Received from task %d...\n",source);
  }
```

Locals | Current Line(s) | Current Stack

Current Line(s)

Name	Value
rank	2104240

Input/... | Break... | Watch... | Stac... | Trace... | Tracepoint ... | Logb...

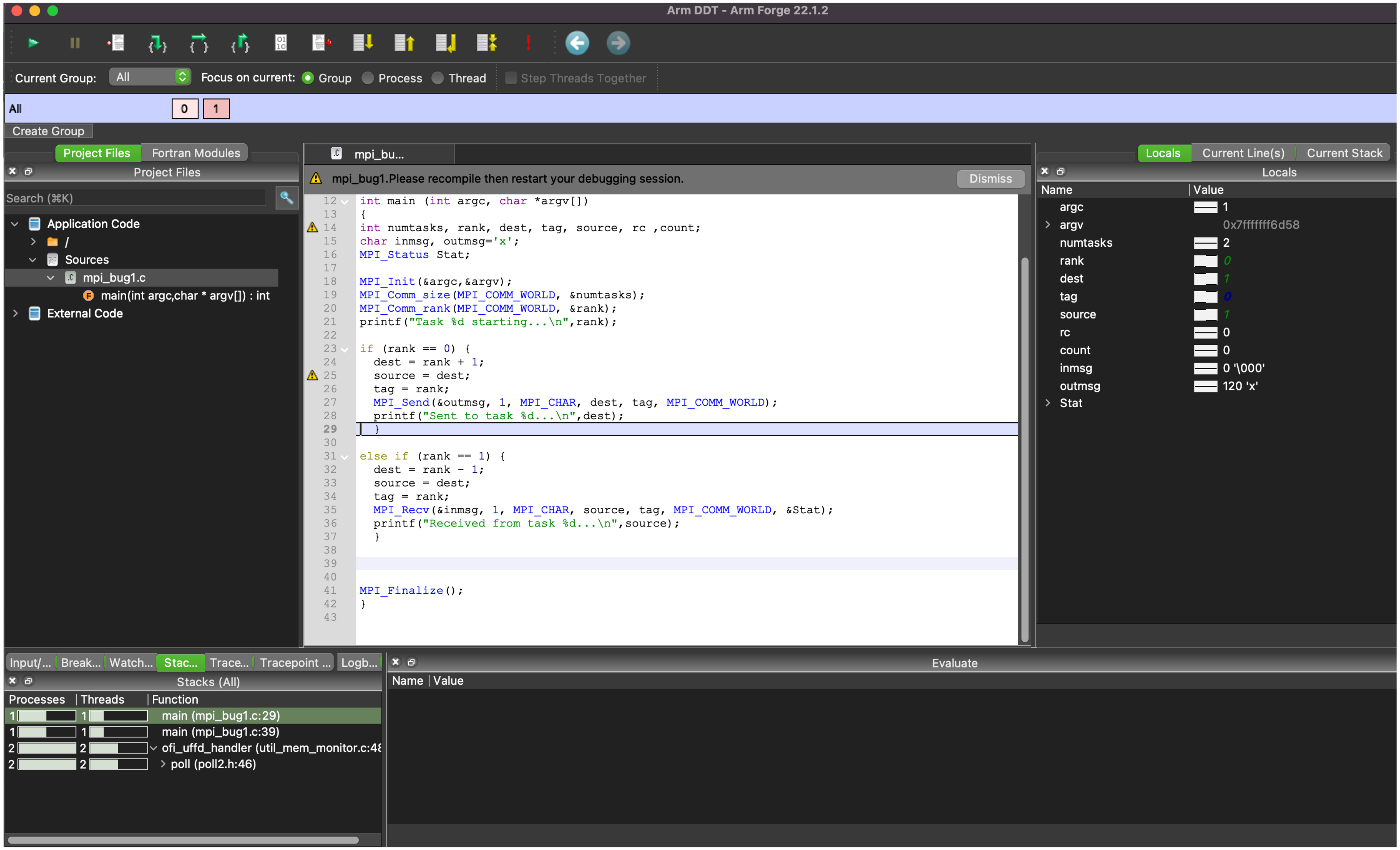
Stacks (All)

Processes	Threads	Function
2	2	main (mpi_bug1.c:20)
2	2	ofi_uffd_handler (util_mem_monitor.c:48
2	2	poll (poll2.h:46)

Evaluate

Name	Value

Step into – local variables Updated



mpi_bug1.Please recompile then restart your debugging session.

```
int main (int argc, char *argv[])
{
int numtasks, rank, dest, tag, source, rc ,count;
char inmsg, outmsg='x';
MPI_Status Stat;

MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
printf("Task %d starting...\n",rank);

if (rank == 0) {
  dest = rank + 1;
  source = dest;
  tag = rank;
  MPI_Send(&outmsg, 1, MPI_CHAR, dest, tag, MPI_COMM_WORLD);
  printf("Sent to task %d...\n",dest);
  }

else if (rank == 1) {
  dest = rank - 1;
  source = dest;
  tag = rank;
  MPI_Recv(&inmsg, 1, MPI_CHAR, source, tag, MPI_COMM_WORLD, &Stat);
  printf("Received from task %d...\n",source);
  }



MPI_Finalize();
}
```

Locals

Name	Value
argc	1
argv	0x7fffffff6d58
numtasks	2
rank	0
dest	1
tag	0
source	1
rc	0
count	0
inmsg	0 '\000'
outmsg	120 'x'
Stat	

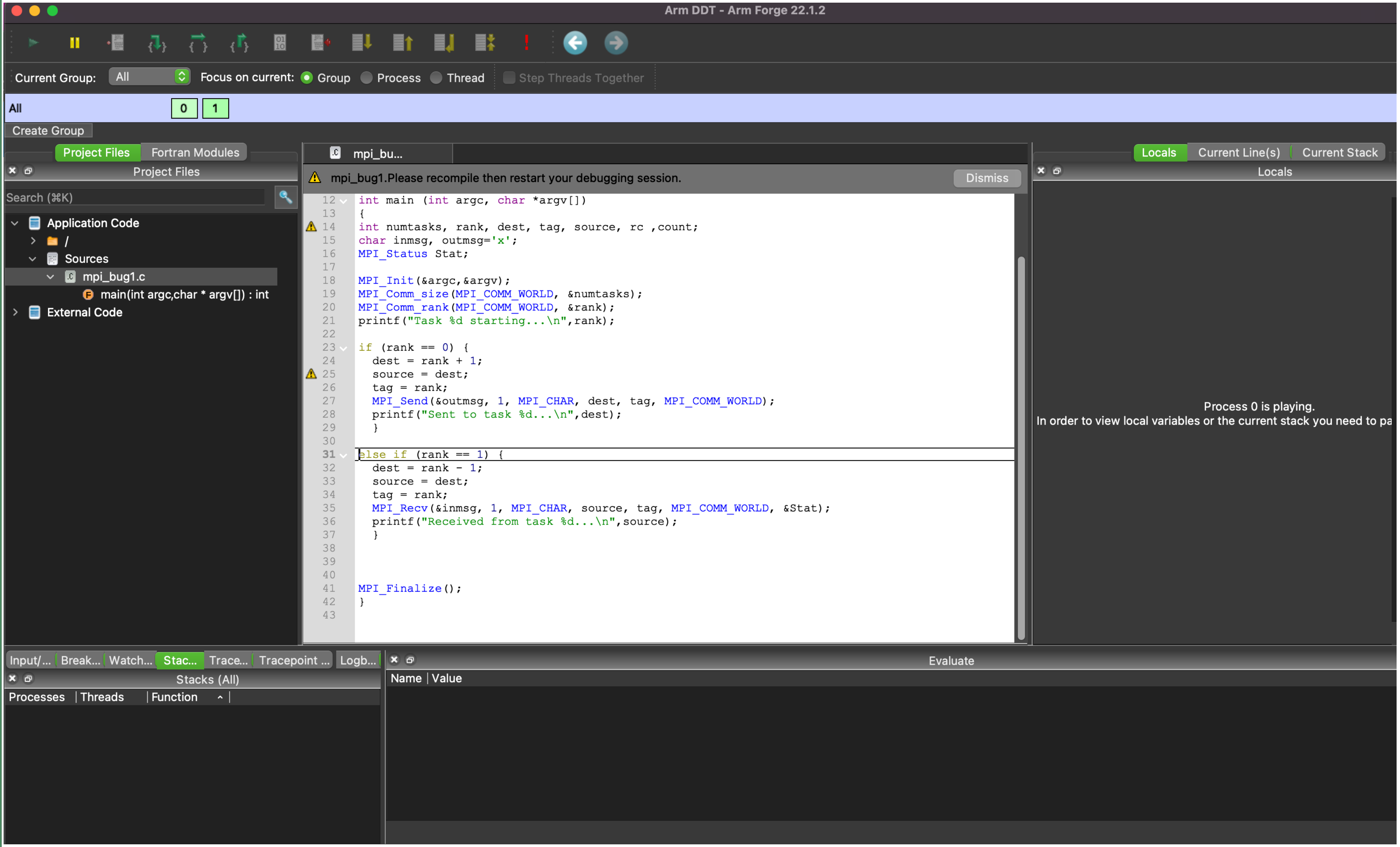
Stacks (All)

Processes	Threads	Function
1	1	main (mpi_bug1.c:29)
1	1	main (mpi_bug1.c:39)
2	2	ofi_uffd_handler (util_mem_monitor.c:48
2	2	poll (poll2.h:46)

OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

Open slide master to edit

Step into – Program is Hanging


Arm DDT - Arm Forge 22.1.2
Current Group: All
Focus on current: Group Process Thread
Step Threads Together
All 0 1
Create Group
Project Files Fortran Modules
Project Files
Search (⌘K)
Application Code
/
Sources
mpi_bug1.c
main(int argc,char * argv[]) : int
External Code
mpi_bu...
Dismiss
Locals Current Line(s) Current Stack
Locals
Input/... Break... Watch... Stac... Trace... Tracepoint ... Logb...
Stacks (All)
Processes Threads Function
Evaluate
Name Value

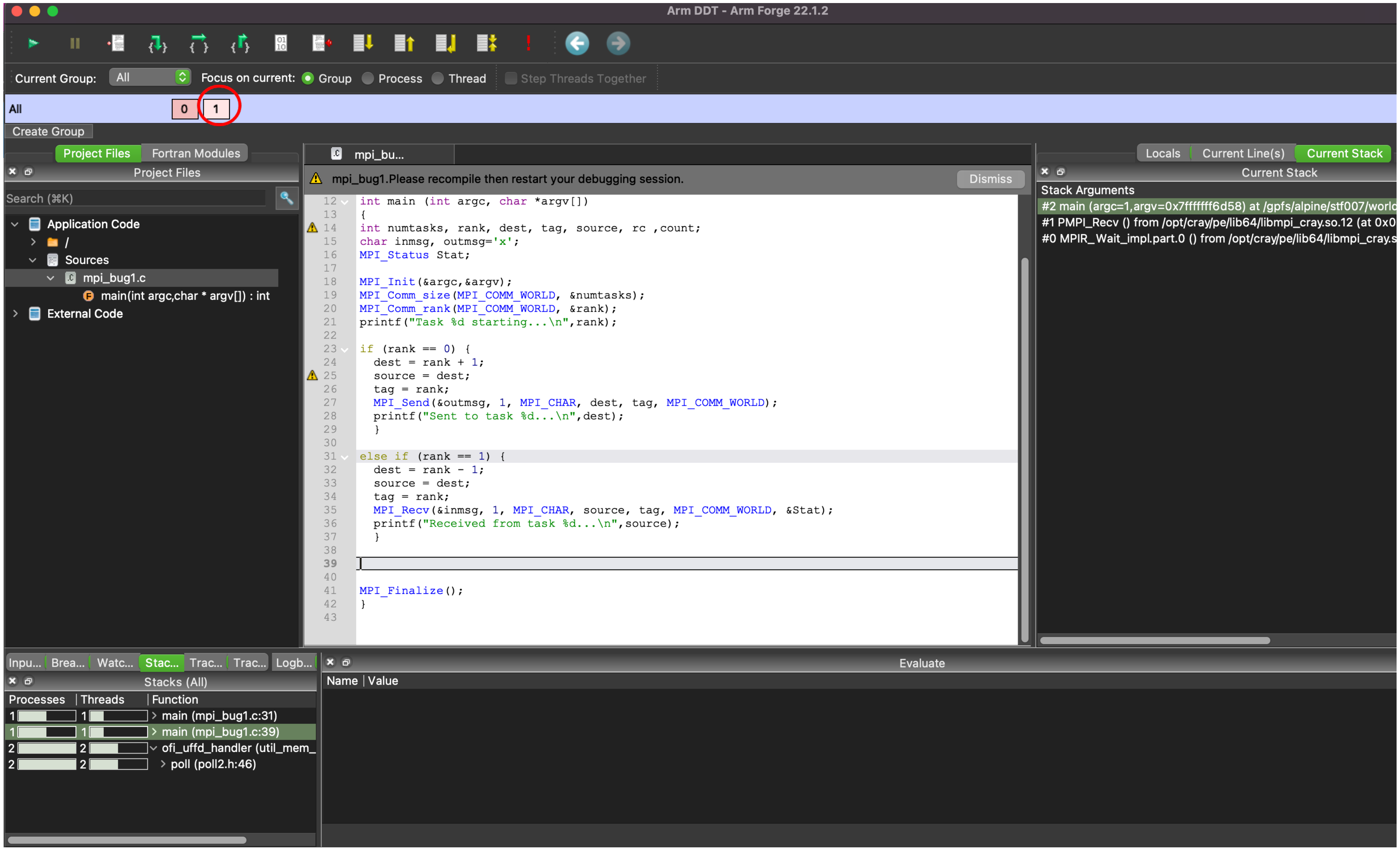
mpi_bug1.Please recompile then restart your debugging session.

```
int main (int argc, char *argv[])
{
int numtasks, rank, dest, tag, source, rc ,count;
char inmsg, outmsg='x';
MPI_Status Stat;

MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
printf("Task %d starting...\n",rank);

if (rank == 0) {
  dest = rank + 1;
  source = dest;
  tag = rank;
  MPI_Send(&outmsg, 1, MPI_CHAR, dest, tag, MPI_COMM_WORLD);
  printf("Sent to task %d...\n",dest);
  }

else if (rank == 1) {
  dest = rank - 1;
  source = dest;
  tag = rank;
  MPI_Recv(&inmsg, 1, MPI_CHAR, source, tag, MPI_COMM_WORLD, &Stat);
  printf("Received from task %d...\n",source);
  }



MPI_Finalize();
}
```

Process 0 is playing.
In order to view local variables or the current stack you need to pa

Open slide master to edit

Step into – MPI_WAIT on Rank 1



mpi_bug1.Please recompile then restart your debugging session.

```
12  int main (int argc, char *argv[])
13  {
14  int numtasks, rank, dest, tag, source, rc ,count;
15  char inmsg, outmsg='x';
16  MPI_Status Stat;
17
18  MPI_Init(&argc,&argv);
19  MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
20  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
21  printf("Task %d starting...\n",rank);
22
23  if (rank == 0) {
24    dest = rank + 1;
25    source = dest;
26    tag = rank;
27    MPI_Send(&outmsg, 1, MPI_CHAR, dest, tag, MPI_COMM_WORLD);
28    printf("Sent to task %d...\n",dest);
29    }
30
31  else if (rank == 1) {
32    dest = rank - 1;
33    source = dest;
34    tag = rank;
35    MPI_Recv(&inmsg, 1, MPI_CHAR, source, tag, MPI_COMM_WORLD, &Stat);
36    printf("Received from task %d...\n",source);
37    }
38
39
40
41  MPI_Finalize();
42  }
43
```

Current Stack

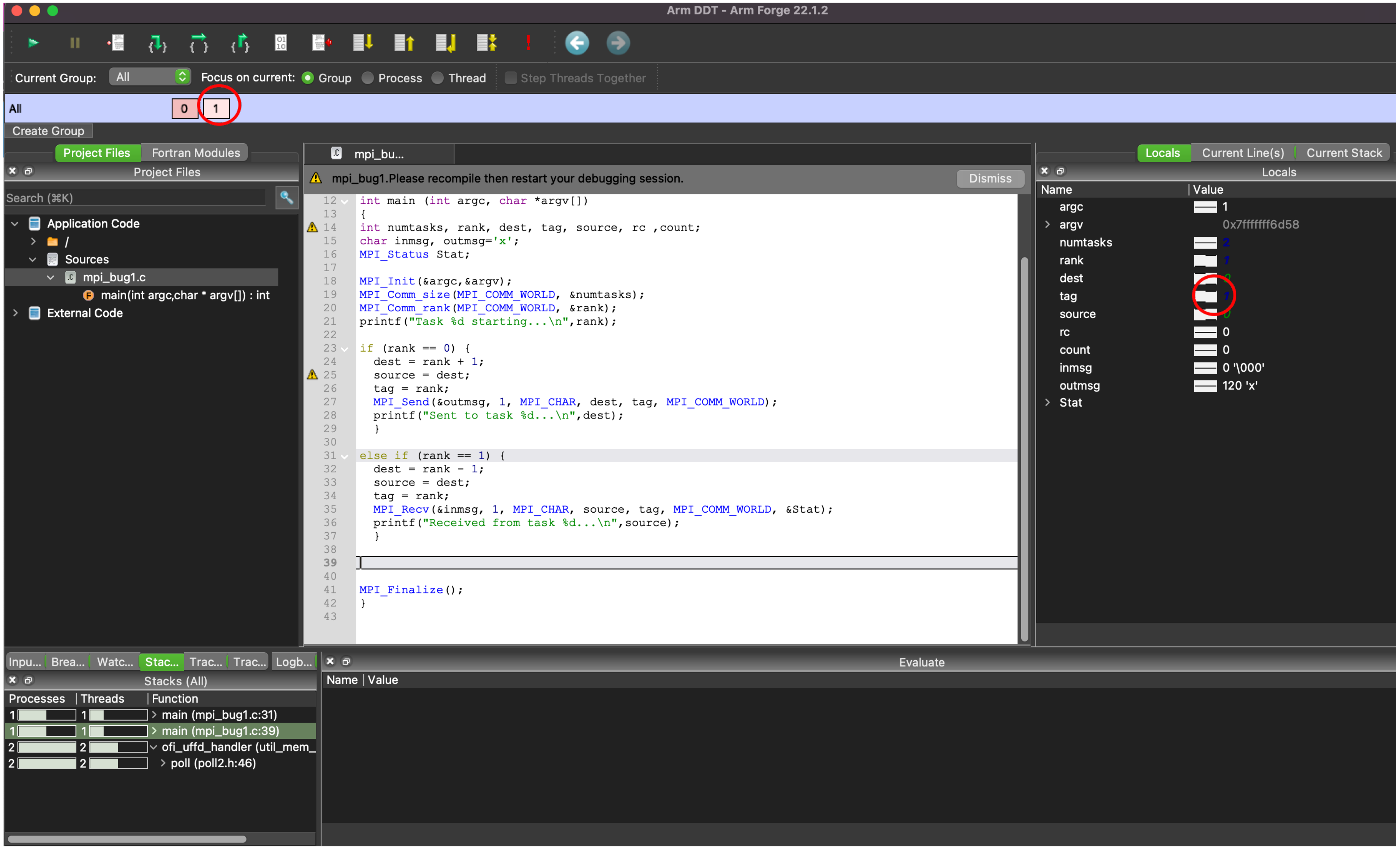
Stack Arguments
#2 main (argc=1,argv=0x7fffffff6d58) at /gpfs/alpine/stf007/world
#1 PMPI_Recv () from /opt/cray/pe/lib64/libmpi_cray.so.12 (at 0x0
#0 MPIR_Wait_impl.part.0 () from /opt/cray/pe/lib64/libmpi_cray.s

Stacks (All)

Processes	Threads	Function
1	1	main (mpi_bug1.c:31)
1	1	main (mpi_bug1.c:39)
2	2	ofi_uffd_handler (util_mem_
2	2	poll (poll2.h:46)

OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

Step into – ‘tag’ variable =1



Arm DDT - Arm Forge 22.1.2

Current Group: All   Focus on current: [✓ Group] Process Thread   Step Threads Together

All   0   1

mpi_bug1.Please recompile then restart your debugging session.   Dismiss

```
12  int main (int argc, char *argv[])
13  {
14  int numtasks, rank, dest, tag, source, rc ,count;
15  char inmsg, outmsg='x';
16  MPI_Status Stat;
17
18  MPI_Init(&argc,&argv);
19  MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
20  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
21  printf("Task %d starting...\n",rank);
22
23  if (rank == 0) {
24    dest = rank + 1;
25    source = dest;
26    tag = rank;
27    MPI_Send(&outmsg, 1, MPI_CHAR, dest, tag, MPI_COMM_WORLD);
28    printf("Sent to task %d...\n",dest);
29    }
30
31  else if (rank == 1) {
32    dest = rank - 1;
33    source = dest;
34    tag = rank;
35    MPI_Recv(&inmsg, 1, MPI_CHAR, source, tag, MPI_COMM_WORLD, &Stat);
36    printf("Received from task %d...\n",source);
37    }
38
39
40
41  MPI_Finalize();
42  }
43
```

Locals

Name	Value
argc	1
argv	0x7fffffff6d58
numtasks	2
rank	1
dest	0
tag	1
source	0
rc	0
count	0
inmsg	0 '\000'
outmsg	120 'x'
Stat	

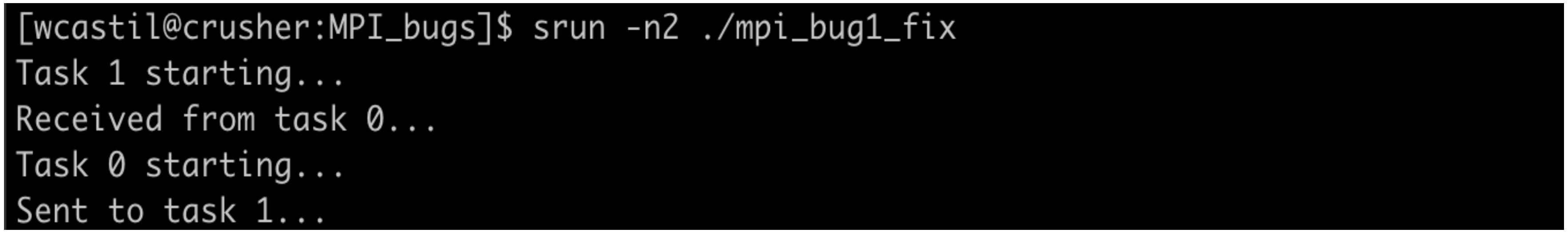
Stacks (All)

Processes	Threads	Function
1	1	main (mpi_bug1.c:31)
1	1	main (mpi_bug1.c:39)
2	2	ofi_uffd_handler (util_mem_
2	2	poll (poll2.h:46)

OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

Open slide master to edit

Correcting the ‘tag’ = 0 for both ranks, program completes



```
[wcastil@crusher:MPI_bugs]$ srun -n2 ./mpi_bug1_fix
Task 1 starting...
Received from task 0...
Task 0 starting...
Sent to task 1...
```