Advanced C++ Course

Linaro Forge

CSCS

Introduction

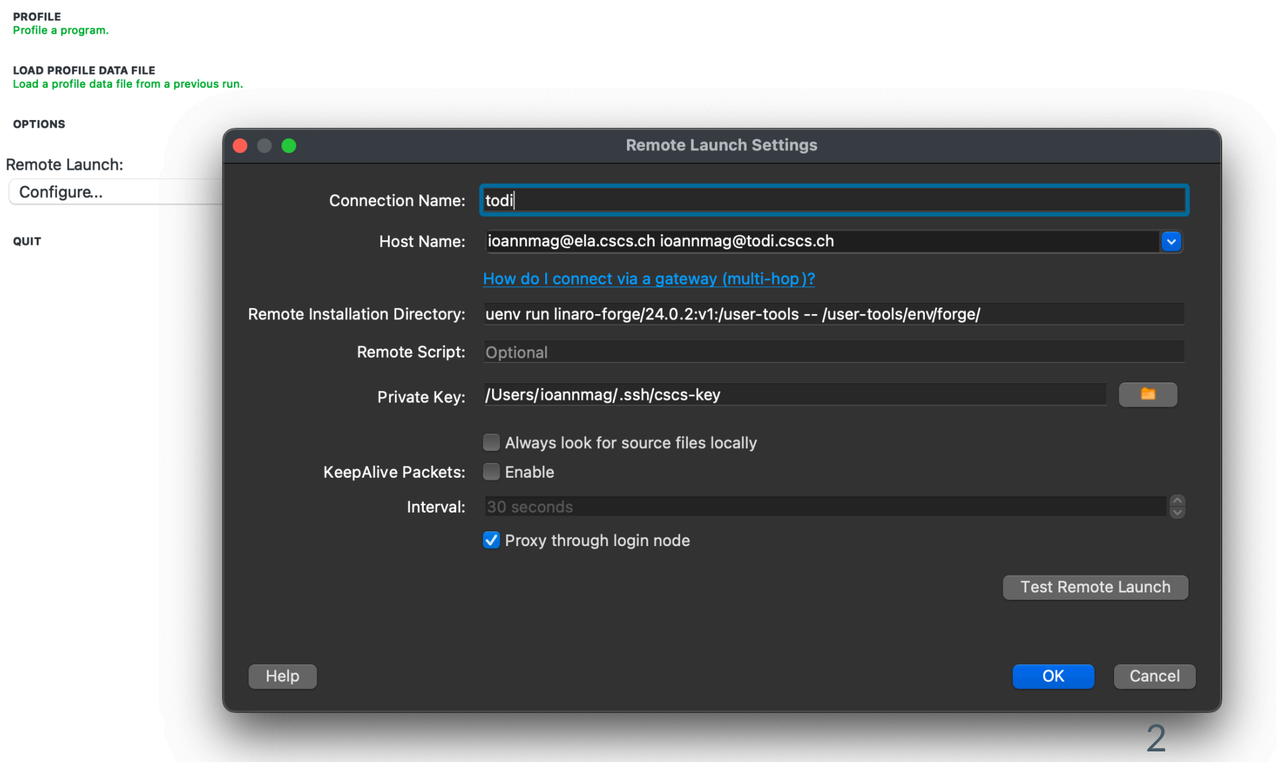
Linaro Forge tools

- MAP
  - "Linaro MAP is a high-performance profiler designed to optimize the efficiency of software running on multicore processors."
  - Sampling profiler
  - Designed for 'hot-spot' analysis with stack traces and performance metrics
  - Supports C/C++/Fortran/Python
  - AMD & Nvidia GPU profiling
  - Performance Reports
- DDT
  - "Linaro DDT is an advanced debugger designed to simplify the troubleshooting and optimization of complex, high-performance computing (HPC) applications."
  - Supports C/C++/Fortran/Python
  - AMD & Nvidia GPUs

Forge client configuration

- Forge client on local machine (Linux, MacOS)
- License
  - Max processes 272 / Max users 256

`TODI` remote setup



MAP

Instructions

Todi

```
uenv start --view=linaro:forge,prgenv-gnu:default prgenv-gnu/24.7 linaro-forge/24.0.2
map -n 4 --mpi=slurm --mpiargs="-A csstaff -p debug -t 10" --profile ./matrix_mult 4
```

Local machine

- Connect to `todi`
- `LOAD PROFILE DATA FILE`

DDT

Instructions

Local machine

- Connect to `todi`
- `MANUAL LAUNCH (ADVANCED)`
- Create correct listener

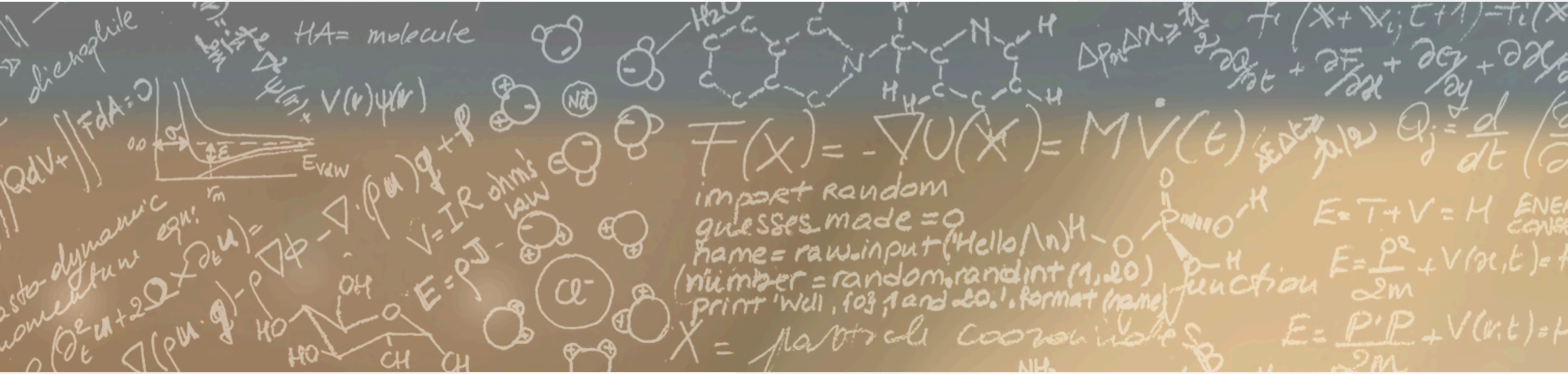
Todi

```
uenv start --view=linaro:forge,prgenv-gnu:default prgenv-gnu/24.7 linaro-forge/24.0.2
srun -n 4 -A csstaff -t30 -p debug ddt-client ./matrix_mult 4
```

Demo

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Questions?

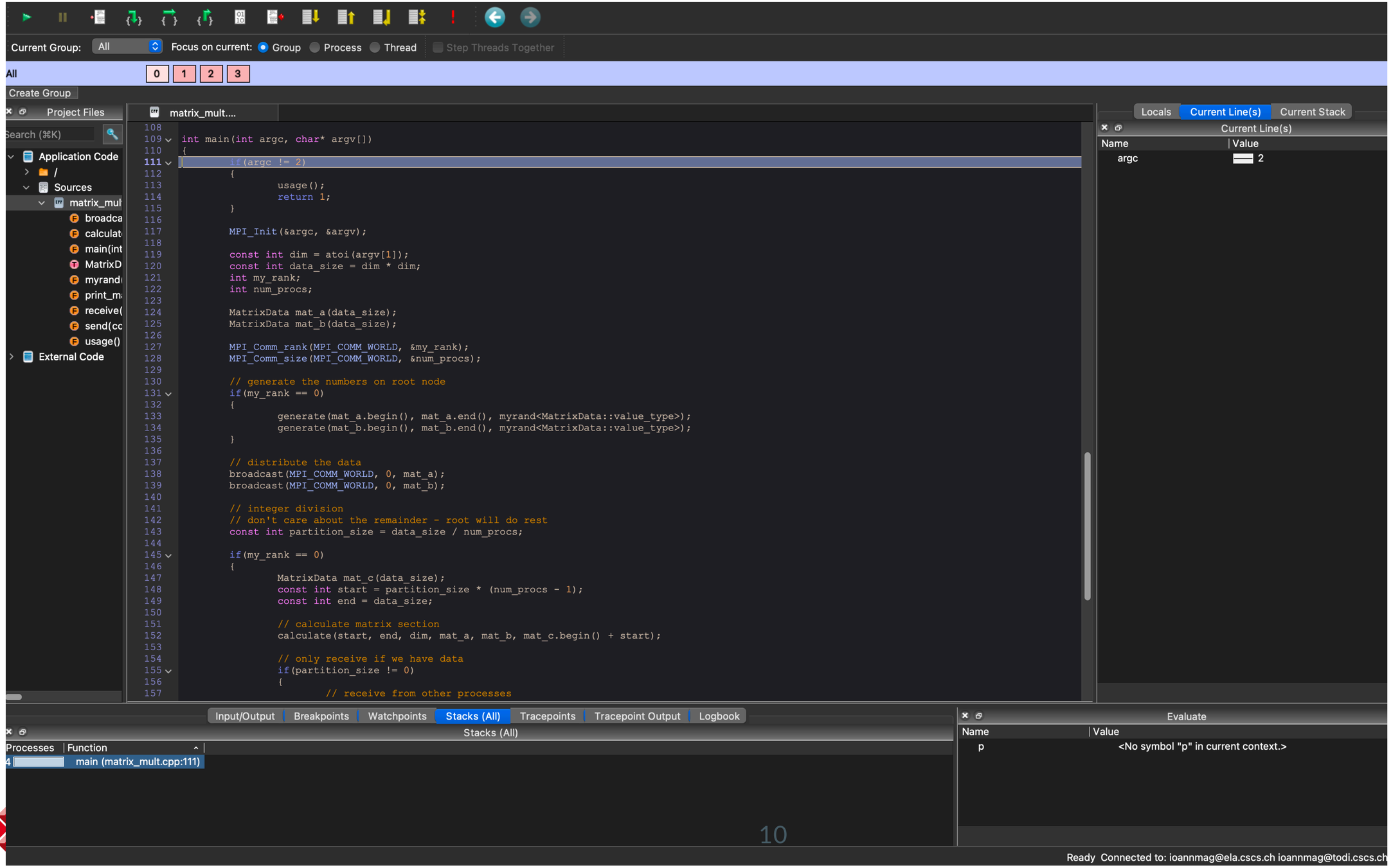
Resources

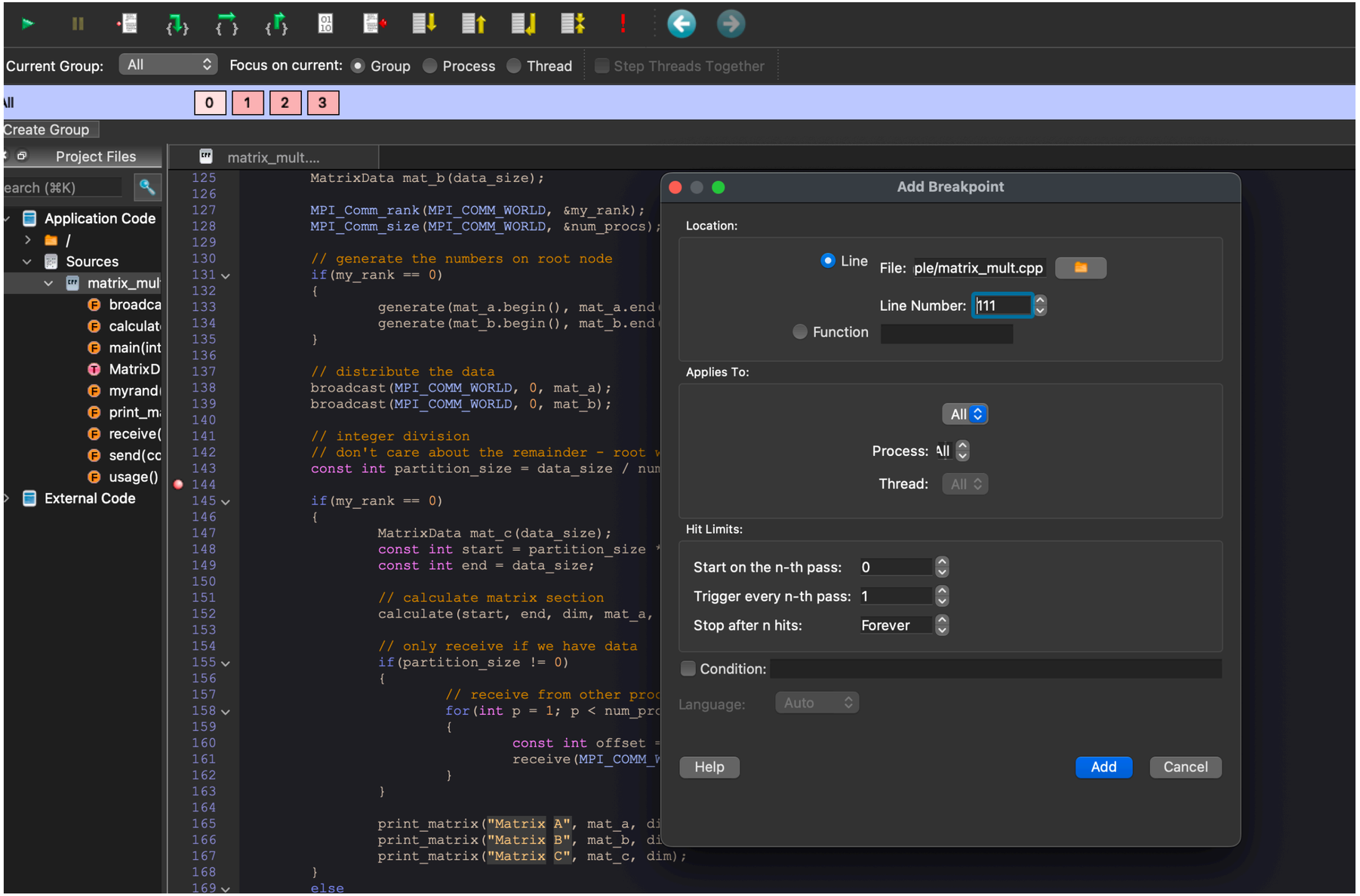
- Linaro MAP website
- Linaro DDT website
- Ensuring Program Correctness with Linaro DDT Rudy Shand

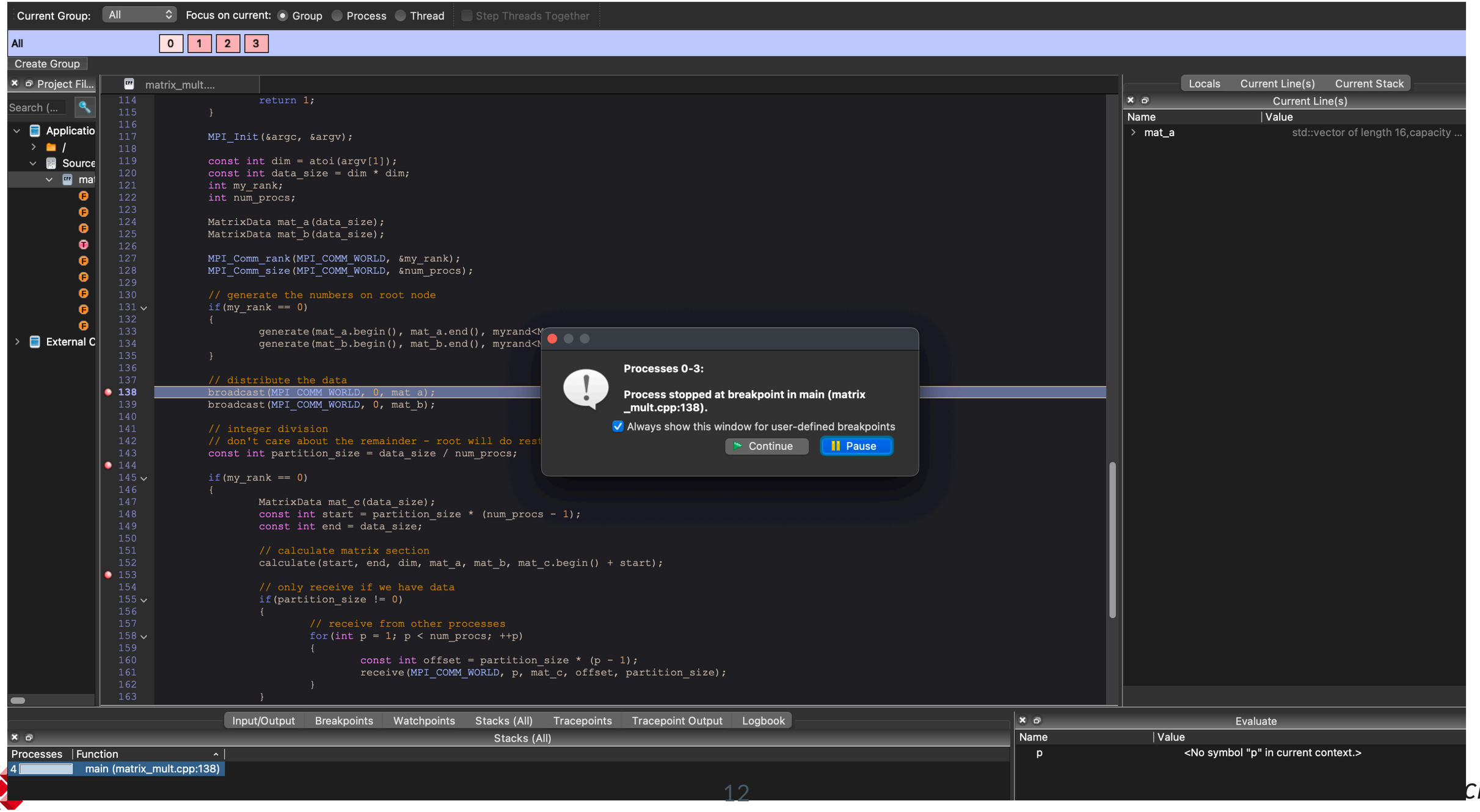
Backup slides

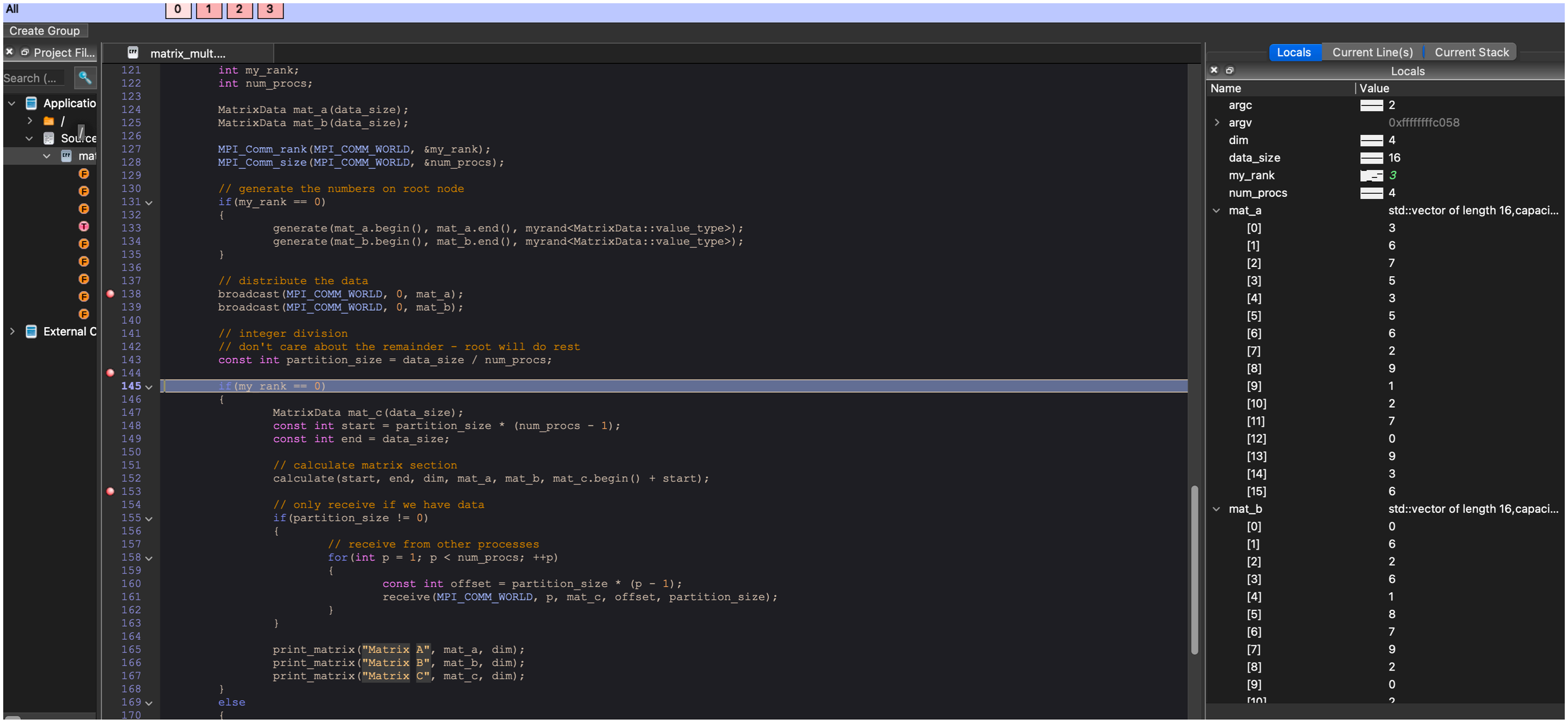
DDT


Linaro DDT - Linaro Forge 24.0.3
linaro
forg
Runtime: manual launch
Details
Number of Processes: 4
CUDA
Details
ROCm
Details...
Intel Xe
Details...
Memory Debugging: Balanced,No guard pages,Backtraces,Preload
Details...
Plugins: none
Details
Help
Options
Listen
Cancel
Get trial licence
Support
linaroforge.com
Linaro Forge 24.0.3 Connected to: ioannmag@ela.cscs.ch ioannmag@todi.cscs.ch

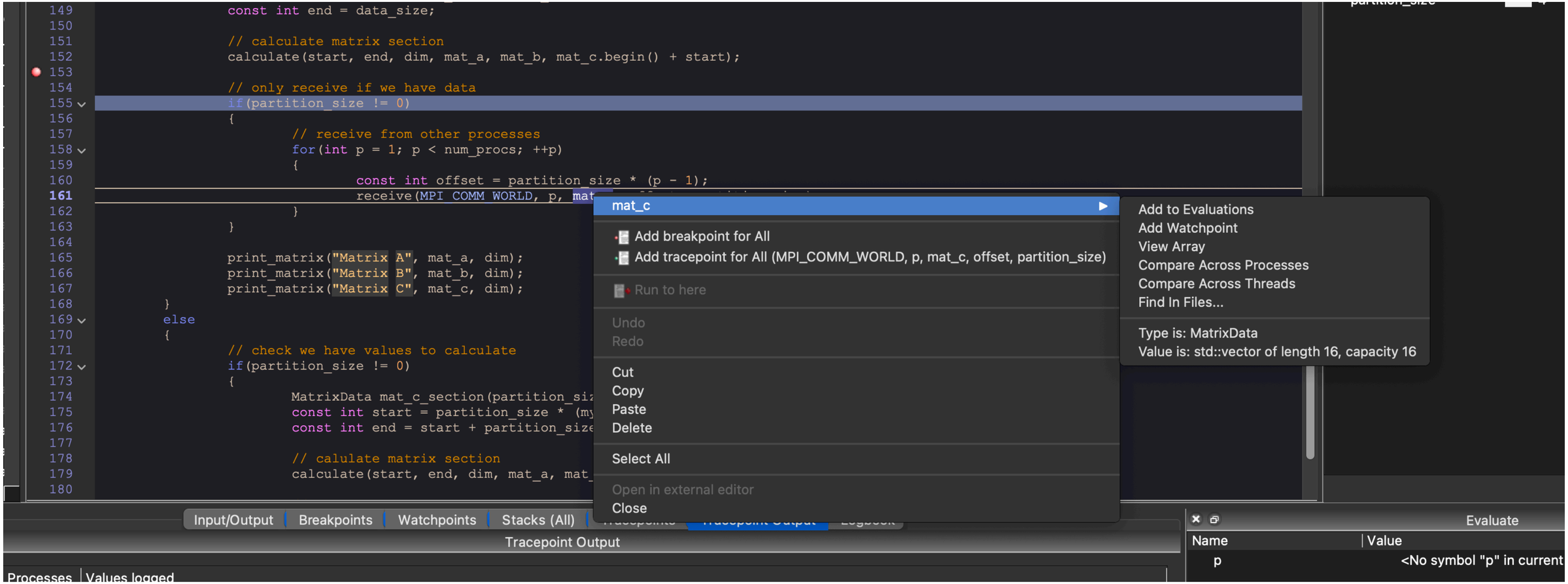

Current Group: All
Focus on current: Group Process Thread
Step Threads Together
All 0 1 2 3
Create Group
Project Files
matrix_mult....
Locals Current Line(s) Current Stack
Current Line(s)
Name Value
argc 2
Application Code
Sources
External Code
int main(int argc, char* argv[])
{
if(argc != 2)
{
usage();
return 1;
}
MPI_Init(&argc, &argv);
const int dim = atoi(argv[1]);
const int data_size = dim * dim;
int my_rank;
int num_procs;
MatrixData mat_a(data_size);
MatrixData mat_b(data_size);
MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
MPI_Comm_size(MPI_COMM_WORLD, &num_procs);
// generate the numbers on root node
if(my_rank == 0)
{
generate(mat_a.begin(), mat_a.end(), myrand<MatrixData::value_type>);
generate(mat_b.begin(), mat_b.end(), myrand<MatrixData::value_type>);
}
// distribute the data
broadcast(MPI_COMM_WORLD, 0, mat_a);
broadcast(MPI_COMM_WORLD, 0, mat_b);
// integer division
// don't care about the remainder - root will do rest
const int partition_size = data_size / num_procs;
if(my_rank == 0)
{
MatrixData mat_c(data_size);
const int start = partition_size * (num_procs - 1);
const int end = data_size;
// calculate matrix section
calculate(start, end, dim, mat_a, mat_b, mat_c.begin() + start);
// only receive if we have data
if(partition_size != 0)
{
// receive from other processes
Input/Output Breakpoints Watchpoints Stacks (All) Tracepoints Tracepoint Output Logbook
Stacks (All)
Processes Function
4 main (matrix_mult.cpp:111)
Evaluate
Name Value
p <No symbol "p" in current context.>
Ready Connected to: ioannmag@ela.cscs.ch ioannmag@todi.cscs.ch

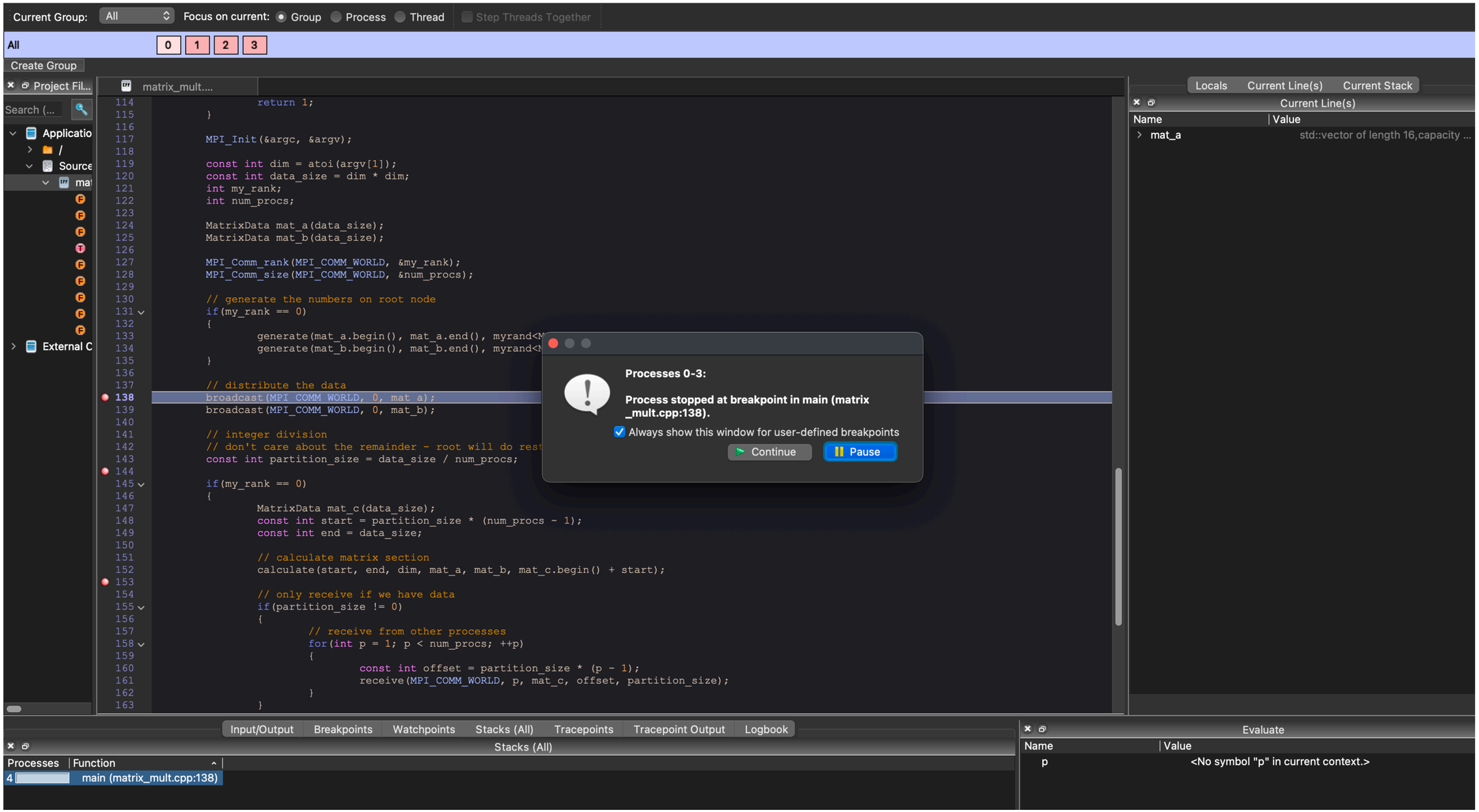

Current Group: All
Focus on current: Group Process Thread
Step Threads Together
0 1 2 3
Create Group
Project Files
matrix_mult....
Application Code
Sources
External Code
MatrixData mat_b(data_size);
MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
MPI_Comm_size(MPI_COMM_WORLD, &num_procs);
// generate the numbers on root node
if(my_rank == 0)
// distribute the data
broadcast(MPI_COMM_WORLD, 0, mat_a);
broadcast(MPI_COMM_WORLD, 0, mat_b);
// integer division
// calculate matrix section
// only receive if we have data
print_matrix("Matrix A", mat_a,
print_matrix("Matrix B", mat_b,
print_matrix("Matrix C", mat_c, dim);
else
Add Breakpoint
Location:
Line
File: ple/matrix_mult.cpp
Line Number: 111
Function
Applies To:
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:
Language: Auto
Help
Add
Cancel

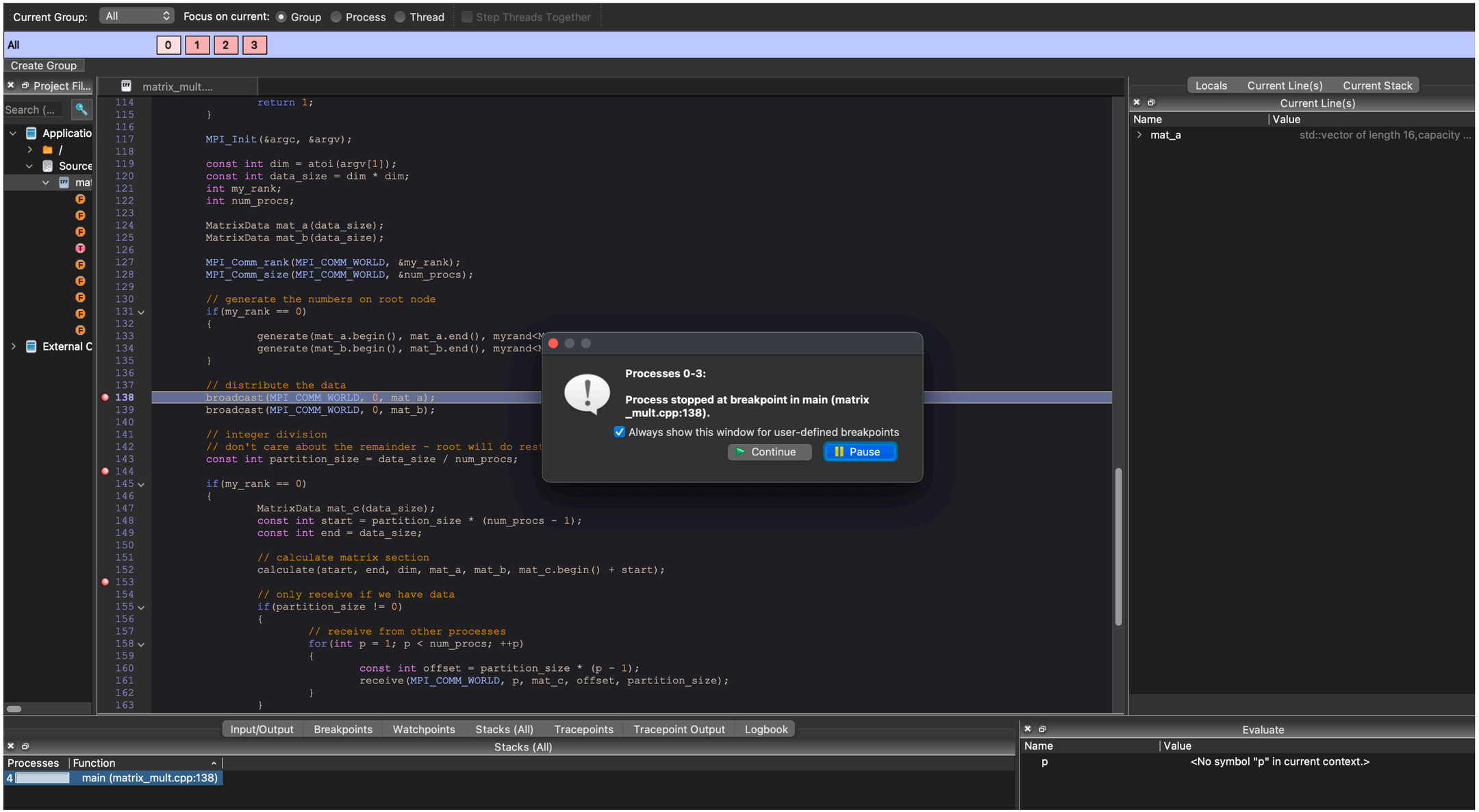

Current Group: All
Focus on current: Group Process Thread
Step Threads Together
All 0 1 2 3
Create Group
Project Fil...
matrix_mult....
Search (...
Applicatio
Source
External C
Locals
Current Line(s)
Current Stack
Name
Value
mat_a
std::vector of length 16,capacity ...
Processes 0-3:
Process stopped at breakpoint in main (matrix _mult.cpp:138).
Always show this window for user-defined breakpoints
Continue
Pause
Input/Output
Breakpoints
Watchpoints
Stacks (All)
Tracepoints
Tracepoint Output
Logbook
Stacks (All)
Processes
Function
4
main (matrix_mult.cpp:138)
Evaluate
Name
Value
p
<No symbol "p" in current context.>

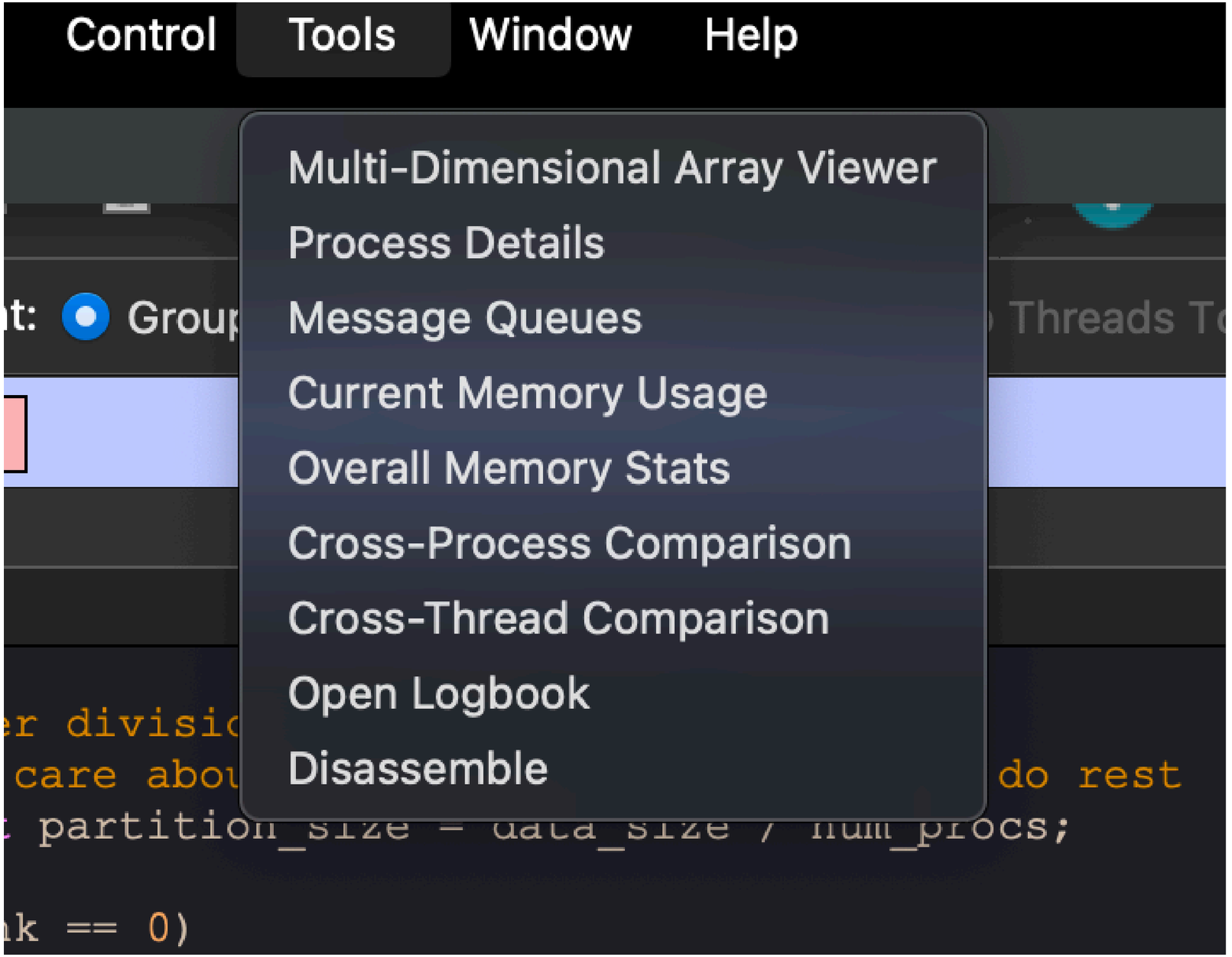


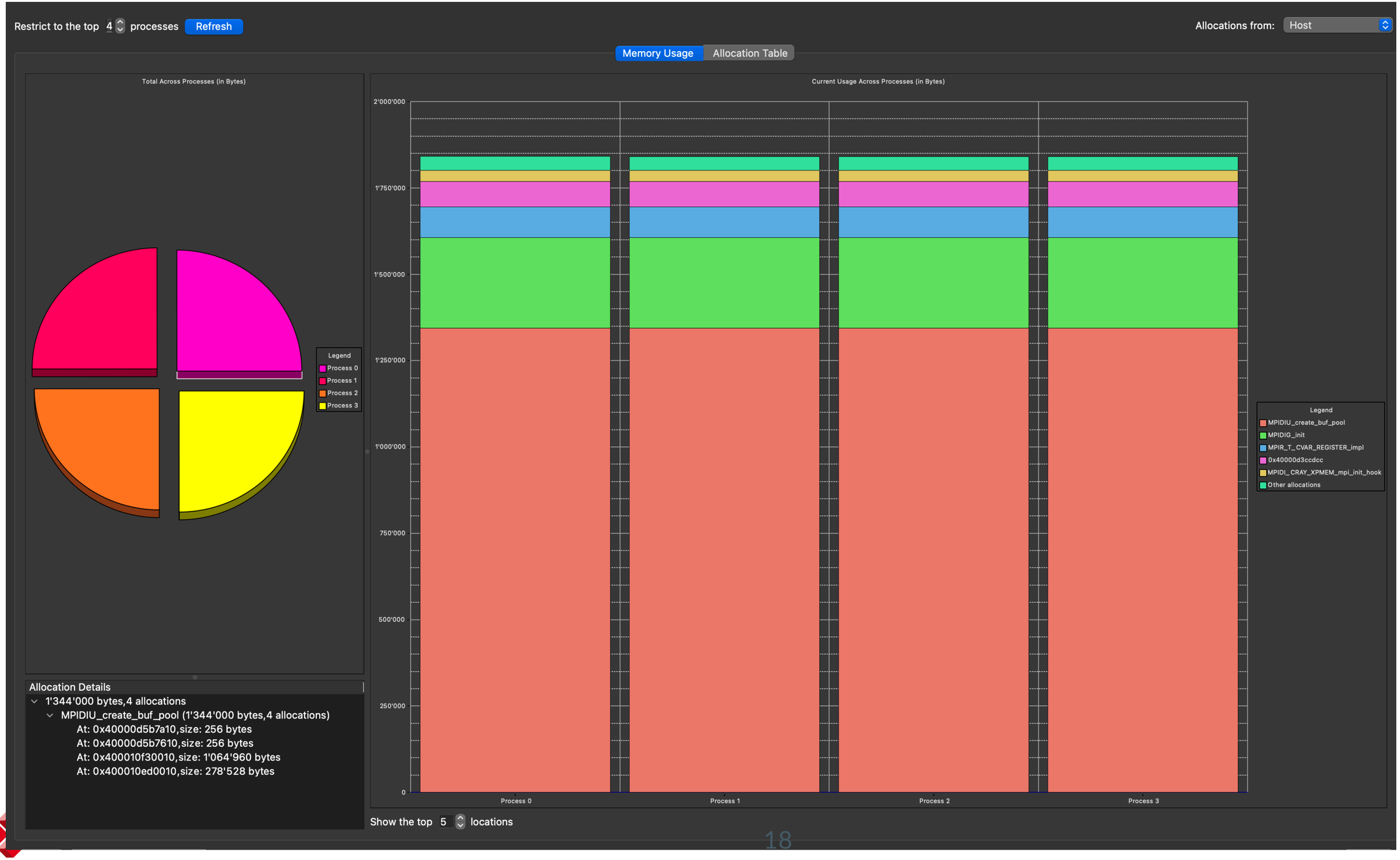

```
121         int my_rank;
122         int num_procs;
123
124         MatrixData mat_a(data_size);
125         MatrixData mat_b(data_size);
126
127         MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
128         MPI_Comm_size(MPI_COMM_WORLD, &num_procs);
129
130         // generate the numbers on root node
131         if(my_rank == 0)
132         {
133                 generate(mat_a.begin(), mat_a.end(), myrand<MatrixData::value_type>);
134                 generate(mat_b.begin(), mat_b.end(), myrand<MatrixData::value_type>);
135         }
136
137         // distribute the data
138         broadcast(MPI_COMM_WORLD, 0, mat_a);
139         broadcast(MPI_COMM_WORLD, 0, mat_b);
140
141         // integer division
142         // don't care about the remainder - root will do rest
143         const int partition_size = data_size / num_procs;
144
145         if(my_rank == 0)
146         {
147                 MatrixData mat_c(data_size);
148                 const int start = partition_size * (num_procs - 1);
149                 const int end = data_size;
150
151                 // calculate matrix section
152                 calculate(start, end, dim, mat_a, mat_b, mat_c.begin() + start);
153
154                 // only receive if we have data
155                 if(partition_size != 0)
156                 {
157                         // receive from other processes
158                         for(int p = 1; p < num_procs; ++p)
159                         {
160                                 const int offset = partition_size * (p - 1);
161                                 receive(MPI_COMM_WORLD, p, mat_c, offset, partition_size);
162                         }
163                 }
164
165                 print_matrix("Matrix A", mat_a, dim);
166                 print_matrix("Matrix B", mat_b, dim);
167                 print_matrix("Matrix C", mat_c, dim);
168         }
169         else
170         {
```


const int end = data_size;
// calculate matrix section
calculate(start, end, dim, mat_a, mat_b, mat_c.begin() + start);
// only receive if we have data
if(partition_size != 0)
{
// receive from other processes
for(int p = 1; p < num_procs; ++p)
{
const int offset = partition_size * (p - 1);
receive(MPI_COMM_WORLD, p, mat
}
}
print_matrix("Matrix A", mat_a, dim);
print_matrix("Matrix B", mat_b, dim);
print_matrix("Matrix C", mat_c, dim);
}
else
{
// check we have values to calculate
if(partition_size != 0)
{
MatrixData mat_c_section(partition_si
const int start = partition_size * (my
const int end = start + partition_size
// calulate matrix section
calculate(start, end, dim, mat_a, mat_
mat_c
Add breakpoint for All
Add tracepoint for All (MPI_COMM_WORLD, p, mat_c, offset, partition_size)
Run to here
Undo
Redo
Cut
Copy
Paste
Delete
Select All
Open in external editor
Close
Add to Evaluations
Add Watchpoint
View Array
Compare Across Processes
Compare Across Threads
Find In Files...
Type is: MatrixData
Value is: std::vector of length 16, capacity 16
Input/Output
Breakpoints
Watchpoints
Stacks (All)
Tracepoint Output
Processes
Values logged
Evaluate
Name
Value
p
<No symbol "p" in current


Current Group: All
Focus on current: Group Process Thread
Step Threads Together
All 0 1 2 3
Create Group
Project Fil...
matrix_mult....
Search (...
Applicatio
Source
External C
Locals
Current Line(s)
Current Stack
Current Line(s)
Name
Value
mat_a
std::vector of length 16,capacity ...
Processes 0-3:
Process stopped at breakpoint in main (matrix _mult.cpp:138).
Always show this window for user-defined breakpoints
Continue
Pause
Input/Output
Breakpoints
Watchpoints
Stacks (All)
Tracepoints
Tracepoint Output
Logbook
Stacks (All)
Processes
Function
4
main (matrix_mult.cpp:138)
Evaluate
Name
Value
p
<No symbol "p" in current context.>


Current Group: All
Focus on current: Group Process Thread
Step Threads Together
All 0 1 2 3
Create Group
Project Fil...
matrix_mult....
Search (...
Applicatio
Source
External C
Locals Current Line(s) Current Stack
Current Line(s)
Name Value
mat_a std::vector of length 16,capacity ...
Processes 0-3:
Process stopped at breakpoint in main (matrix _mult.cpp:138).
Always show this window for user-defined breakpoints
Continue
Pause
Input/Output Breakpoints Watchpoints Stacks (All) Tracepoints Tracepoint Output Logbook
Stacks (All)
Processes Function
4 main (matrix_mult.cpp:138)
Evaluate
Name Value
p <No symbol "p" in current context.>


Control
Tools
Window
Help
Multi-Dimensional Array Viewer
Process Details
Message Queues
Current Memory Usage
Overall Memory Stats
Cross-Process Comparison
Cross-Thread Comparison
Open Logbook
Disassemble


Restrict to the top 4 processes
Refresh
Allocations from: Host
Memory Usage
Allocation Table
Total Across Processes (in Bytes)
Legend
Process 0
Process 1
Process 2
Process 3
Current Usage Across Processes (in Bytes)
2'000'000
1'750'000
1'500'000
1'250'000
1'000'000
750'000
500'000
250'000
0
Process 0
Process 1
Process 2
Process 3
Legend
MPIDIU_create_buf_pool
MPIDIG_init
MPIR_T_CVAR_REGISTER_impl
0x40000d3ccdcc
MPIDI_CRAY_XPMEM_mpi_init_hook
Other allocations
Allocation Details
1'344'000 bytes,4 allocations
MPIDIU_create_buf_pool (1'344'000 bytes,4 allocations)
At: 0x40000d5b7a10,size: 256 bytes
At: 0x40000d5b7610,size: 256 bytes
At: 0x400010f30010,size: 1'064'960 bytes
At: 0x400010ed0010,size: 278'528 bytes
Show the top 5 locations


Graph View
Table View
Total Bytes
Total Calls
Current Bytes
Total bytes allocated/freed
2'000'000
1'750'000
1'500'000
1'250'000
1'000'000
750'000
500'000
250'000
0
Mean (All)
Mean (All)
Process 3
Process 3
Process 1
Process 1
Process 0
Process 0
Process 2
Process 2
Legend
allocation calls
deallocation calls

MAP