



**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich



# Advanced C++ Course

Linaro Forge

CSCS

# Introduction

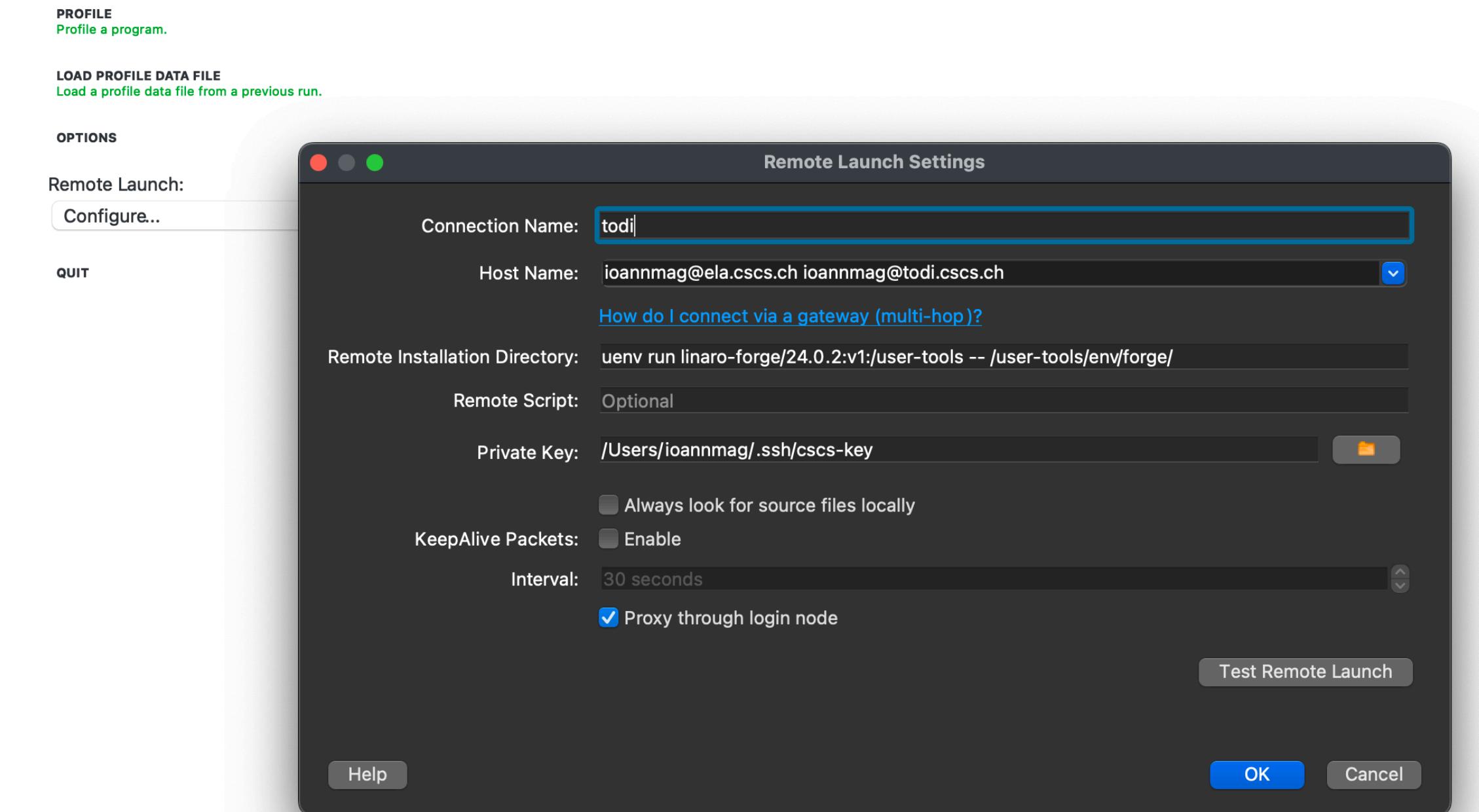
## Linaro Forge tools

- 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)

## TODI remote setup



# MAP

## Instructions

### Todi

```
uenv start --view=linaro:forge,prgen-gnu:default prgen-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
```

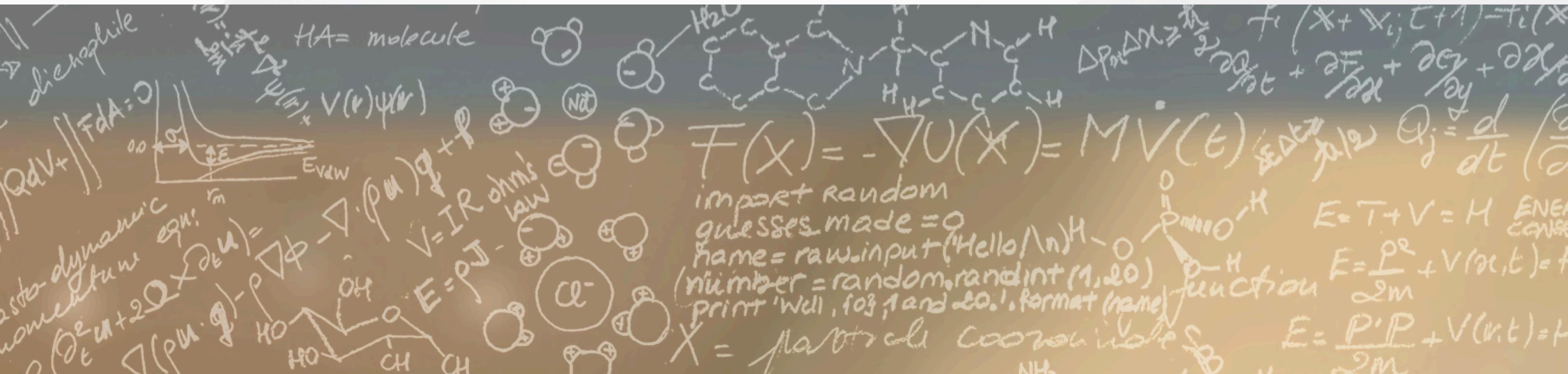
# Demo



CSCS

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

ETH zürich



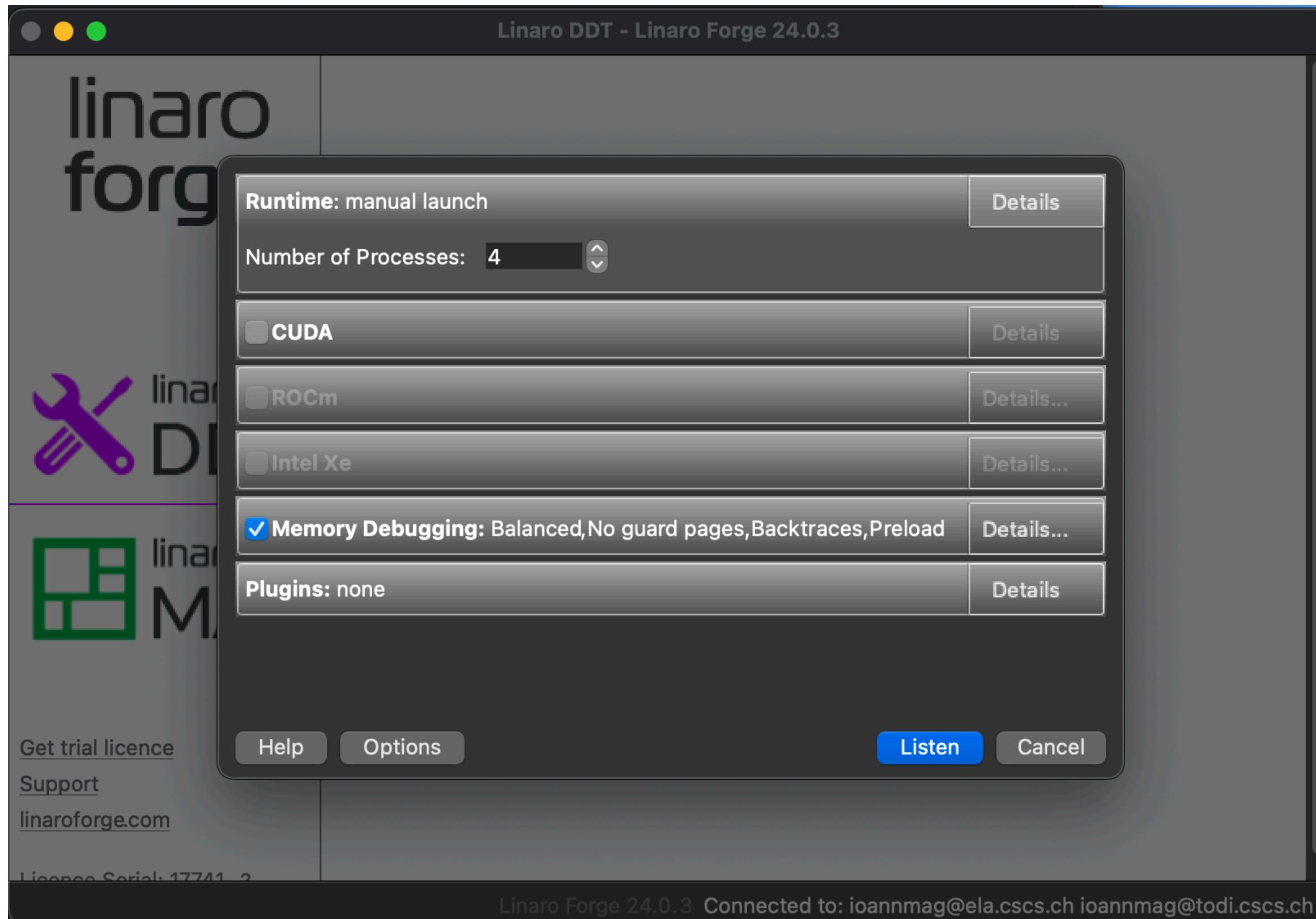
# Questions?

# Resources

- [Linaro MAP website](#)
- [Linaro DDT website](#)
- [Ensuring Program Correctness with Linaro DDT Rudy Shand](#)

# Backup slides

# DDT



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

All 0 1 2 3

Create Group

Project Files matrix\_mult...

Search (⌘K)

Application Code

- /
- Sources
- matrix\_mult...
- broadcast()
- calculate()
- main(int)
- MatrixD
- myrand()
- print\_matrix()
- receive()
- send()
- usage()

External Code

108  
109 int main(int argc, char\* argv[]){  
110 {  
111 if(argc != 2)  
112 {  
113 usage();  
114 return 1;  
115 }  
116 MPI\_Init(&argc, &argv);  
117 const int dim = atoi(argv[1]);  
118 const int data\_size = dim \* dim;  
119 int my\_rank;  
120 int num\_procs;  
121  
122 MatrixData mat\_a(data\_size);  
123 MatrixData mat\_b(data\_size);  
124  
125 MPI\_Comm\_rank(MPI\_COMM\_WORLD, &my\_rank);  
126 MPI\_Comm\_size(MPI\_COMM\_WORLD, &num\_procs);  
127  
128 // generate the numbers on root node  
129 if(my\_rank == 0)  
130 {  
131 generate(mat\_a.begin(), mat\_a.end(), myrand<MatrixData::value\_type>);  
132 generate(mat\_b.begin(), mat\_b.end(), myrand<MatrixData::value\_type>);  
133 }  
134  
135 // distribute the data  
136 broadcast(MPI\_COMM\_WORLD, 0, mat\_a);  
137 broadcast(MPI\_COMM\_WORLD, 0, mat\_b);  
138  
139 // integer division  
140 // don't care about the remainder - root will do rest  
141 const int partition\_size = data\_size / num\_procs;  
142  
143 if(my\_rank == 0)  
144 {  
145 MatrixData mat\_c(data\_size);  
146 const int start = partition\_size \* (num\_procs - 1);  
147 const int end = data\_size;  
148  
149 // calculate matrix section  
150 calculate(start, end, dim, mat\_a, mat\_b, mat\_c.begin() + start);  
151  
152 // only receive if we have data  
153 if(partition\_size != 0)  
154 {  
155 // receive from other processes  
156 }

Locals Current Line(s) Current Stack

Name	Value
argc	2

Input/Output Breakpoints Watchpoints Stacks (All) Tracepoints Tracepoint Output Logbook

Processes Function

4 main (matrix\_mult.cpp:111)

Evaluate

Name	Value
p	<No symbol "p" in current context.>

Ready Connected to: ioannmag@ela.csccs.ch ioannmag@todi.csccs.ch

The screenshot shows a debugger interface with a code editor and a floating 'Add Breakpoint' dialog.

**Code Editor:**

- Title:** matrix\_mult....
- File:** ple/matrix\_mult.cpp
- Line Number:** 111
- Content:** The code implements MPI-based matrix multiplication. It includes MPI operations like MPI\_Comm\_rank, MPI\_Comm\_size, MPI\_Broadcast, and MPI\_Recv. It also includes local operations like generate, calculate, and print\_matrix.

**Add Breakpoint Dialog:**

- Location:**
  - Line: File: ple/matrix\_mult.cpp
  - Line Number: 111
  - Function: [disabled]
- 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:** [disabled]
- Language:** Auto
- Buttons:** Help, Add, Cancel

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

All 0 1 2 3

Create Group

\* Project File... Search (...) Application Source mat External C

```

114         return 1;
115     }
116
117     MPI_Init(&argc, &argv);
118
119     const int dim = atoi(argv[1]);
120     const int data_size = dim * dim;
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<int>());
134         generate(mat_b.begin(), mat_b.end(), myrand<int>());
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     }

```

Processes 0-3:  
Process stopped at breakpoint in main (matrix\_mult.cpp:138).

Always show this window for user-defined breakpoints

Continue | Pause



All    0 | 1 | 2 | 3

Create Group

Project Fil... matrix\_mult....

Search (...

Application / Source mat External C

```

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     {

```

Locals

Name	Value
argc	2
argv	0xfffffffffc058
dim	4
data_size	16
my_rank	3
num_procs	4
mat_a	std::vector of length 16, capaci...
[0]	3
[1]	6
[2]	7
[3]	5
[4]	3
[5]	5
[6]	6
[7]	2
[8]	9
[9]	1
[10]	2
[11]	7
[12]	0
[13]	9
[14]	3
[15]	6
mat_b	std::vector of length 16, capaci...
[0]	0
[1]	6
[2]	2
[3]	6
[4]	1
[5]	8
[6]	7
[7]	9
[8]	2
[9]	0
r101	?

```
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 {
171     // check we have values to calculate
172     if(partition_size != 0)
173     {
174         MatrixData mat_c_section(partition_size);
175         const int start = partition_size * (my_rank - 1);
176         const int end = start + partition_size;
177
178         // calculate matrix section
179         calculate(start, end, dim, mat_a, mat_b, mat_c_section);
180     }
}

```

partition\_size

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

Type is: MatrixData  
Value is: std::vector of length 16, capacity 16

Input/Output | Breakpoints | Watchpoints | Stacks (All) | Tracepoints | Tracepoint Output | Logbook | 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 File... matrix\_mult....

Search (...

Application / Source matrix\_mult.cpp

External C

```

114         return 1;
115     }
116
117     MPI_Init(&argc, &argv);
118
119     const int dim = atoi(argv[1]);
120     const int data_size = dim * dim;
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<int>());
134         generate(mat_b.begin(), mat_b.end(), myrand<int>());
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     }

```

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.>

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

All 0 1 2 3

Create Group

Project File... matrix\_mult....

Search (...

Application / Source matrix\_mult.cpp

External C

```

114         return 1;
115     }
116
117     MPI_Init(&argc, &argv);
118
119     const int dim = atoi(argv[1]);
120     const int data_size = dim * dim;
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<int>());
134         generate(mat_b.begin(), mat_b.end(), myrand<int>());
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     }

```

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.>

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

gt: ● Group

er division  
care about  
partition\_size = data\_size / num\_procs;

ik == 0)

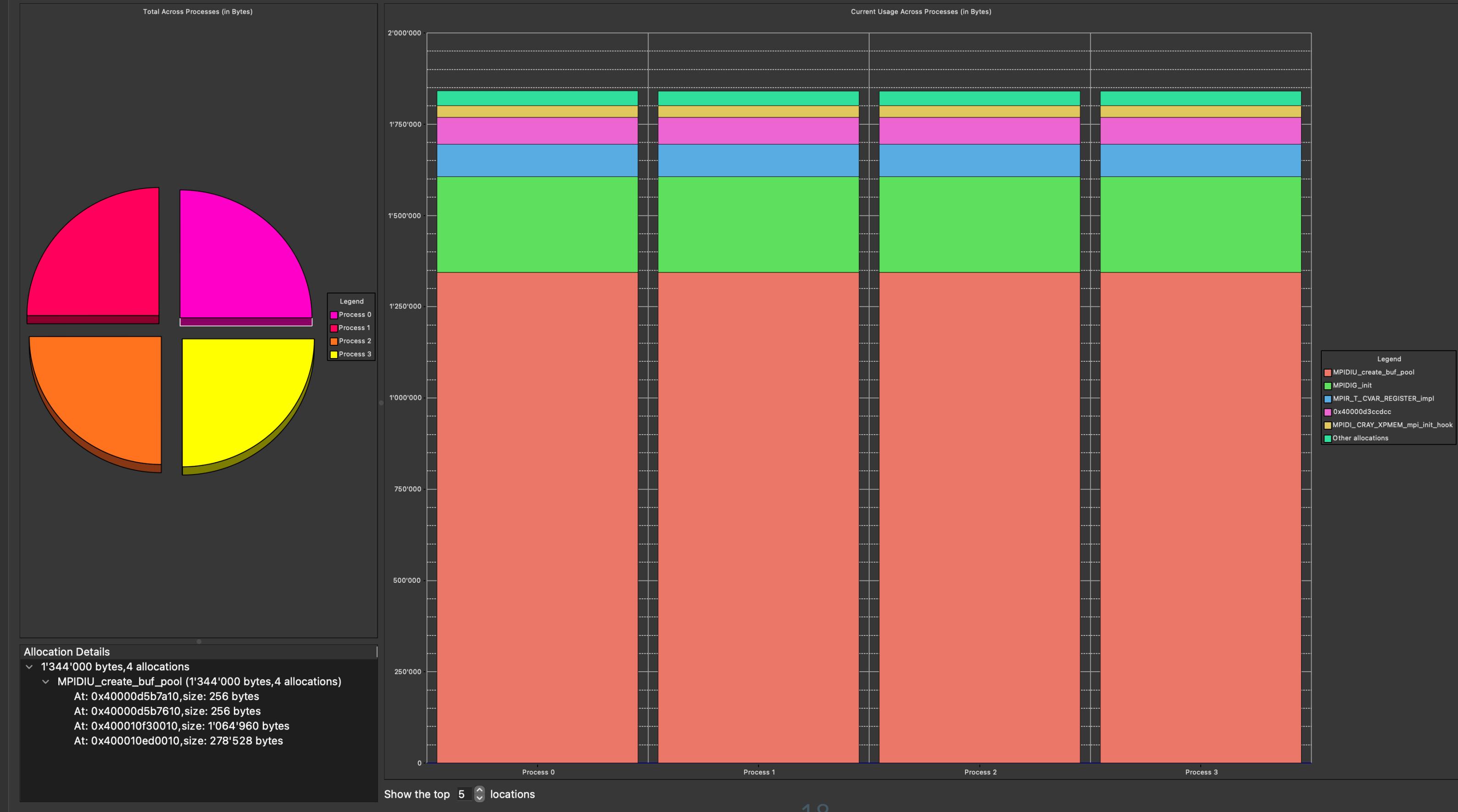
Threads To

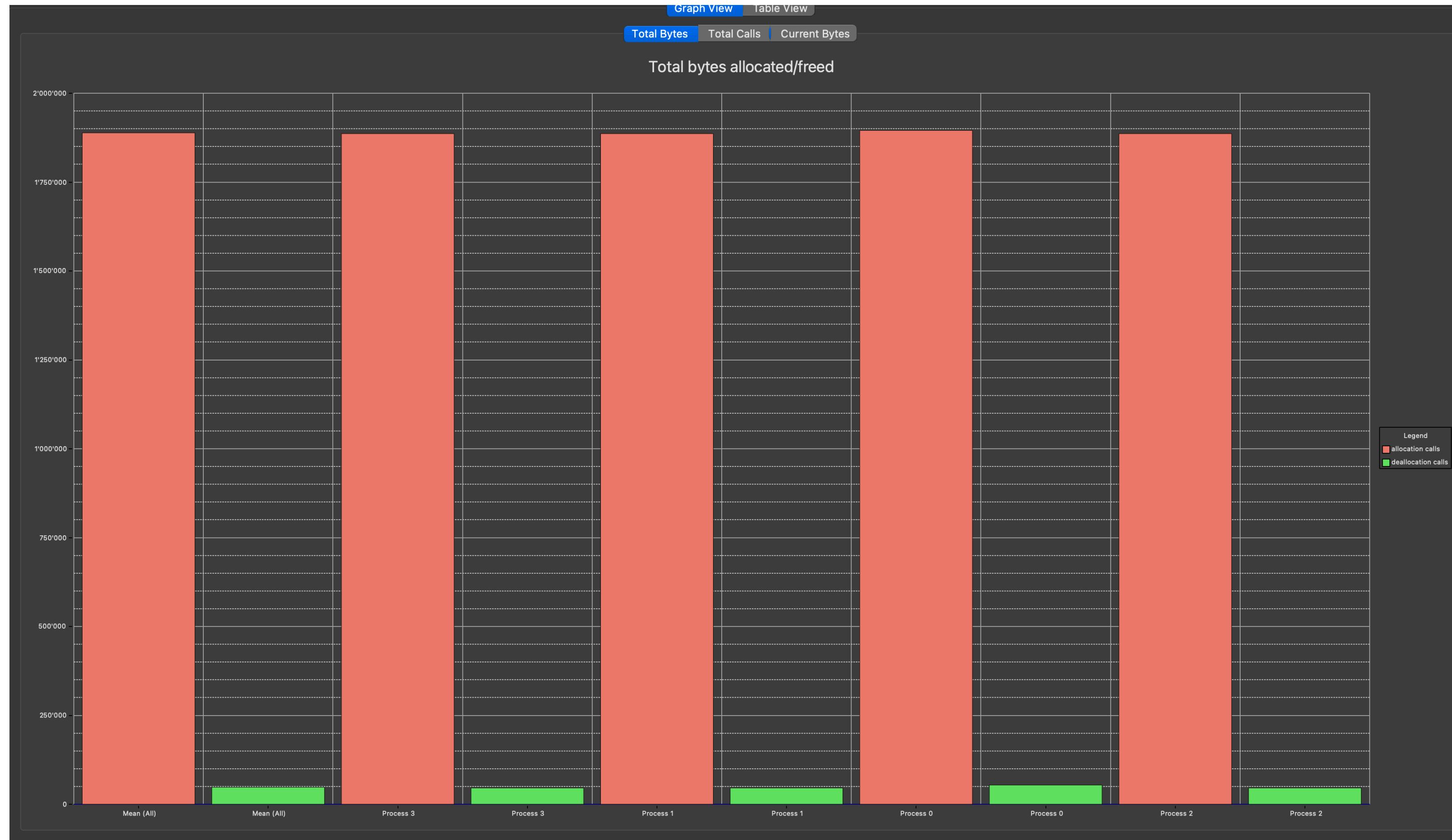
do rest

Restrict to the top 4 processes Refresh

Allocations from: Host

Memory Usage Allocation Table





# MAP