allinea
Now part of ARM

High performance tools to debug, profile, and analyze your applications

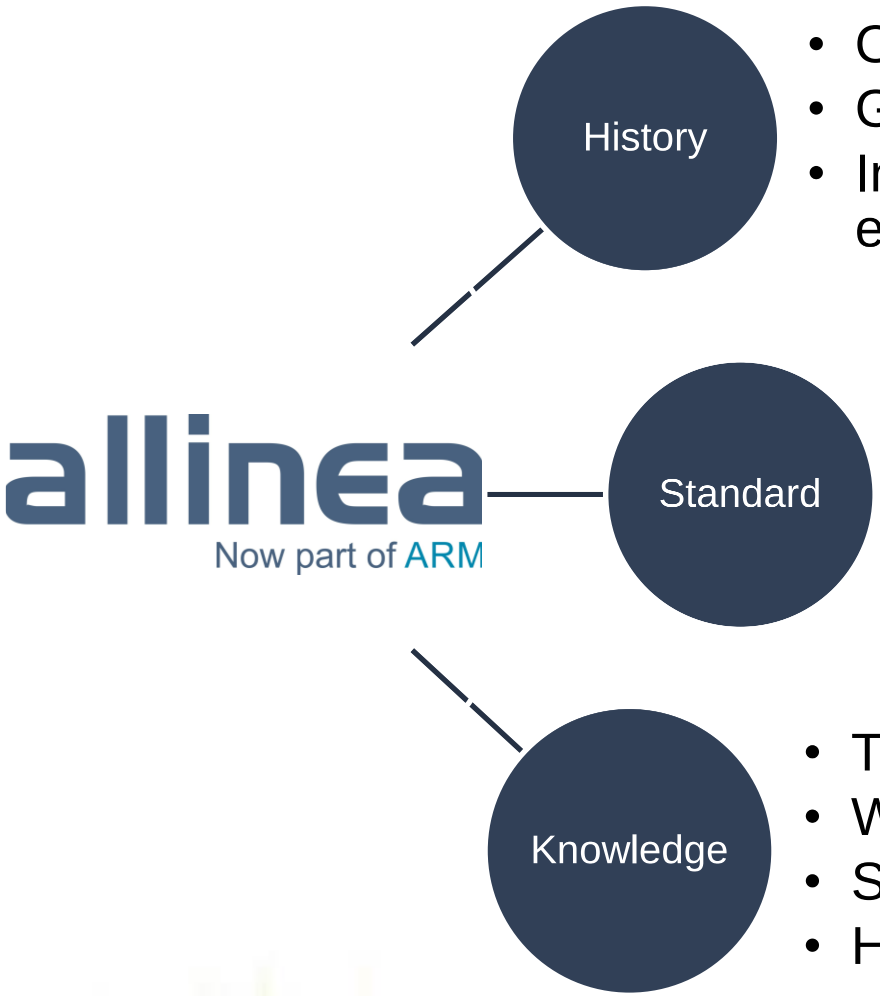
Allinea Tools - ARM

goingARM Workshop – ISC 2017

Oliver Perks (olly.perks@arm.com)

allinea FORGE allinea DDT allinea MAP allinea PERFORMANCE REPORTS

Allinea: Heritage in HPC



- Over 10 years
- Global team
- Industry experience

- 7 of the top 10
- Multi discipline

- Training
- Webinars
- Services
- H2020 research

Overview of Allinea Tools

- Allinea Forge Pro
  - Debugger and profiler
  - Same user interface
  - Deep application insight
  - For application developers

- Allinea Performance Reports
  - Application performance summary
  - For system administrators
  - Historical performance tracking

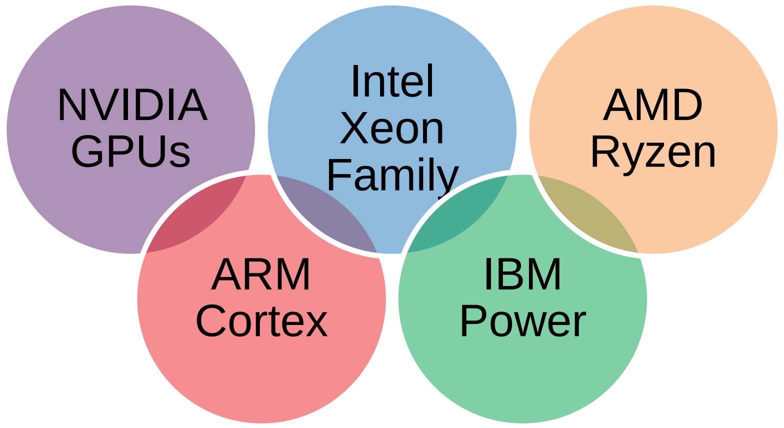
Allinea Now Part of ARM

- Allinea acquired by ARM in December 2016

- What this means for Allinea:
  - Continuation of cross platform product
  - Strength to deliver roadmap faster

- What this means for ARM:
  - Better support for ARM based hardware
  - Step towards a coherent HPC tools ecosystem
  - Wealth of HPC knowledge and experience

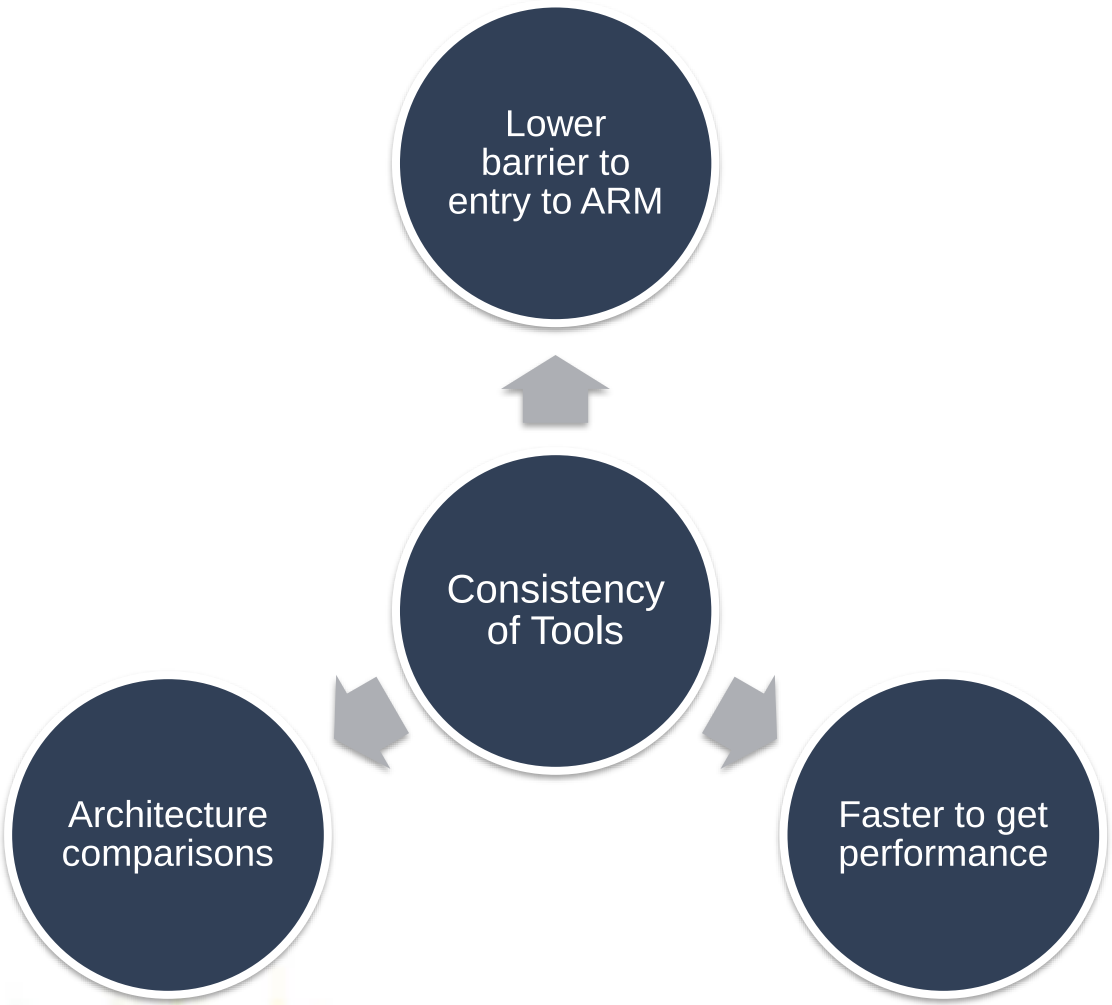
Cross Platform Tools



- Consistent tools
  - Across multiple architectures
  - Same user interface and experience
- Exploiting new features
  - Making the most out of the hardware
    - New CUDA profiling
  - Hardware specific performance metrics

allinea Now part of ARM

Cross Platform: Why does it matter?



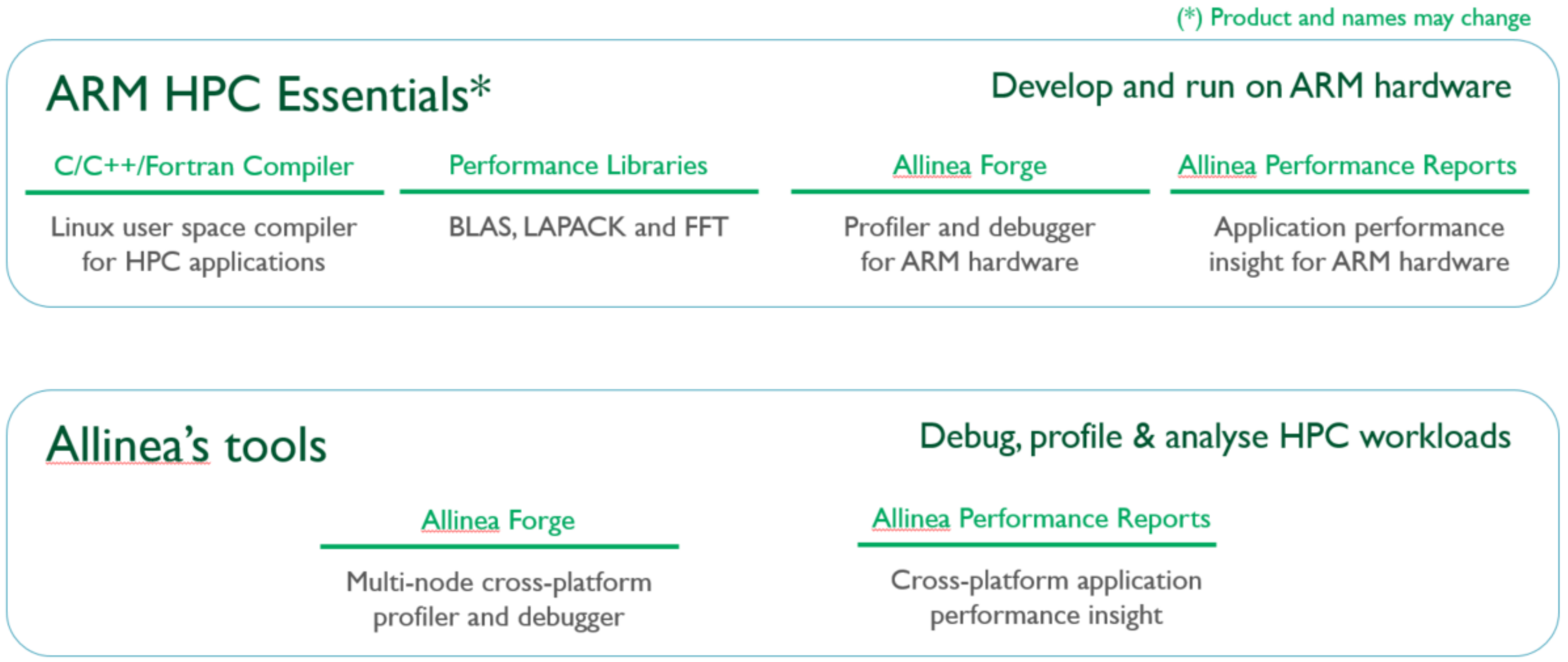
Porting to ARM

- Porting codes: Consistency
  - Debugging code is integral to porting
  - Compare variables & arrays
- Porting codes: Performance
  - Understand performance on old platform
  - Measure and optimise performance on ARM
  - Consistent view aids porting
- Don’t go it alone
  - Training, professional services
  - Other tools
  - Community

allinea Now part of ARM

Ecosystem

- ARM supported HPC tools
  - Allinea is now part of the ARM HPC ecosystem



(*) Product and names may change

ARM HPC Essentials* — Develop and run on ARM hardware

C/C++/Fortran Compiler	Performance Libraries	Allinea Forge	Allinea Performance Reports
Linux user space compiler for HPC applications	BLAS, LAPACK and FFT	Profiler and debugger for ARM hardware	Application performance insight for ARM hardware

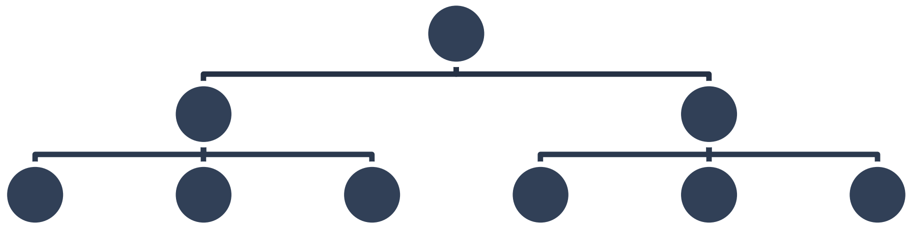
Allinea's tools — Debug, profile & analyse HPC workloads

Allinea Forge	Allinea Performance Reports
Multi-node cross-platform profiler and debugger	Cross-platform application performance insight

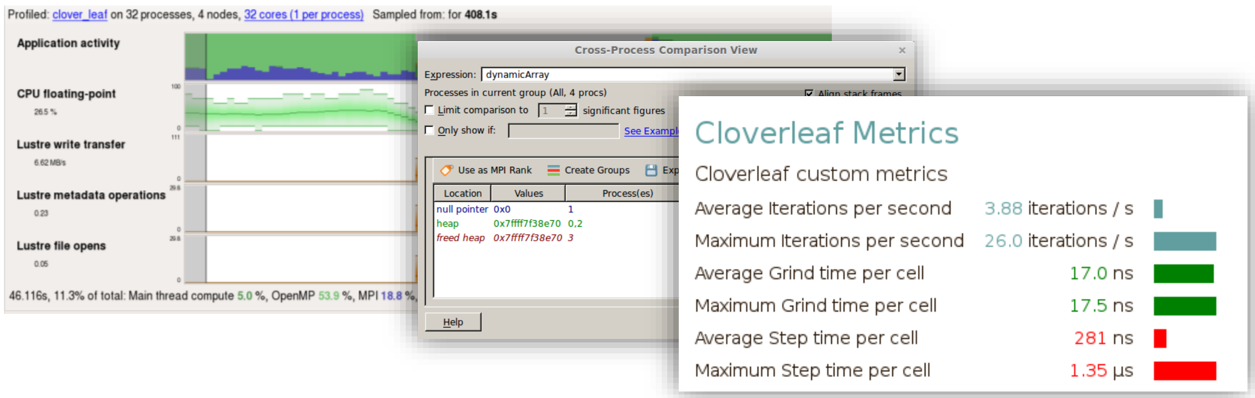
allinea
Now part of ARM

Generating useful and meaningful information

Scalable & Portable



Data collection



Profiled: clover_leaf on 32 processes, 4 nodes, 32 cores (1 per process) Sampled from: for **408.1s**

Application activity
CPU floating-point 26.5 %
Lustre write transfer 6.62 MB/s
Lustre metadata operations 0.23
Lustre file opens 0.05

46.116s, 11.3% of total: Main thread compute 5.0 %, OpenMP 53.9 %, MPI 18.8 %,

Cross-Process Comparison View

Expression: dynamicArray

Processes in current group (All, 4 procs)

☐ Limit comparison to 1 significant figures

☐ Only show if: See Exampl

☑ Align stack frames

Use as MPI Rank
Create Groups
Exp
Help

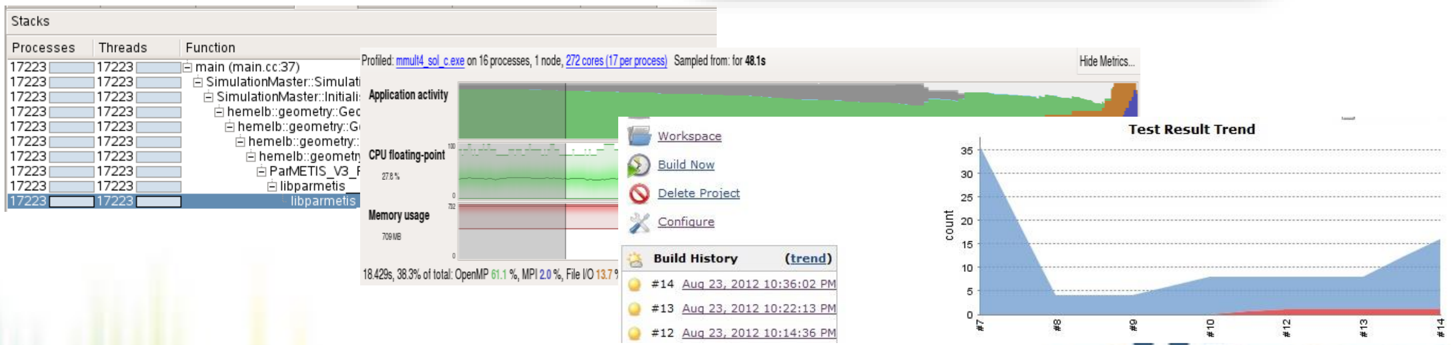
Location	Values	Process(es)
null pointer	0x0	1
heap	0x7ffff7f38e70	0,2
freed heap	0x7ffff7f38e70	3

Cloverleaf Metrics

Cloverleaf custom metrics

Average Iterations per second	3.88 iterations / s	
Maximum Iterations per second	26.0 iterations / s	
Average Grind time per cell	17.0 ns	🟩
Maximum Grind time per cell	17.5 ns	🟩
Average Step time per cell	281 ns	🟥
Maximum Step time per cell	1.35 µs	🟥

Data Presentation



Stacks

```
Processes  Threads  Function
17223      17223    ⊟ main (main.cc:37)
17223      17223      ⊟ SimulationMaster::Simulati
17223      17223        ⊟ SimulationMaster::Initiali
17223      17223          ⊟ hemelb::geometry::Gec
17223      17223            ⊟ hemelb::geometry::G
17223      17223              ⊟ hemelb::geometry::
17223      17223                ⊟ hemelb::geometry
17223      17223                  ⊟ ParMETIS_V3_F
17223      17223                    ⊟ libparmetis
17223      17223                        libparmetis
```

Profiled: mmult4_sol_c.exe on 16 processes, 1 node, 272 cores (17 per process) Sampled from: for **48.1s**

18.429s, 38.3% of total: OpenMP 61.1 %, MPI 2.0 %, File I/O 13.7 %

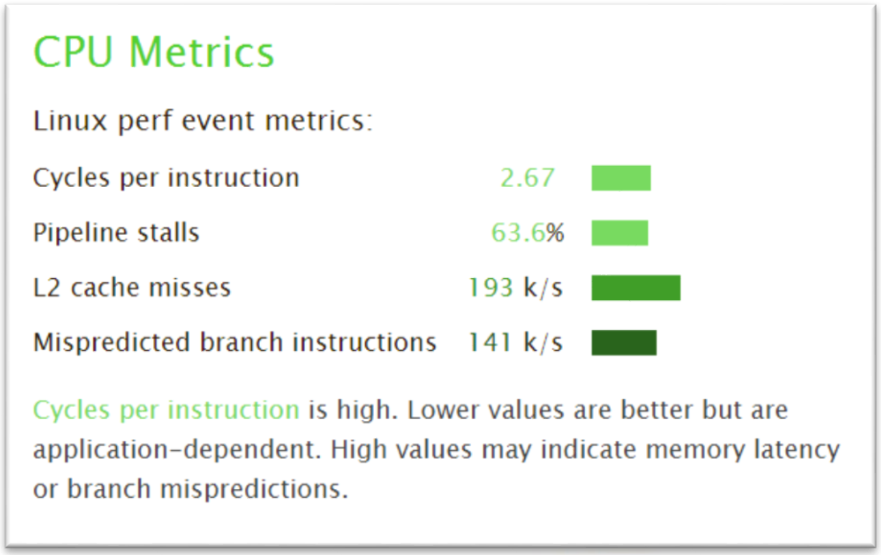
Build History (trend)

#14	Aug 23, 2012 10:36:02 PM
#13	Aug 23, 2012 10:22:13 PM
#12	Aug 23, 2012 10:14:36 PM

allinea Now part of ARM

ARM Support

- DDT debugger support
  - Same functionality as X86
  - No watchpoints
- MAP & Performance Reports
  - Same system metrics
  - No CPU instruction time
  - Using Linux Perf



CPU Metrics

Linux perf event metrics:

Cycles per instruction	2.67	🟩
Pipeline stalls	63.6%	🟩
L2 cache misses	193 k/s	🟩
Mispredicted branch instructions	141 k/s	🟩

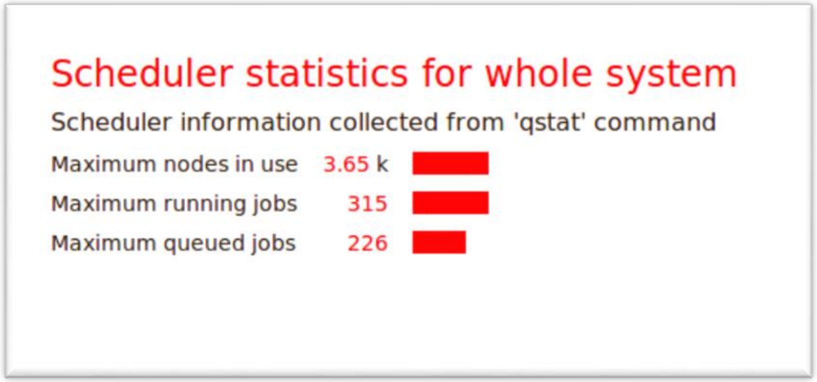
Cycles per instruction is high. Lower values are better but are application-dependent. High values may indicate memory latency or branch mispredictions.

allinea Now part of ARM

Advanced Profiler Usage

- Custom Metrics
  - Platforms can differ
  - Capture platform specific data
  - Using MAP framework
  - Application specific data

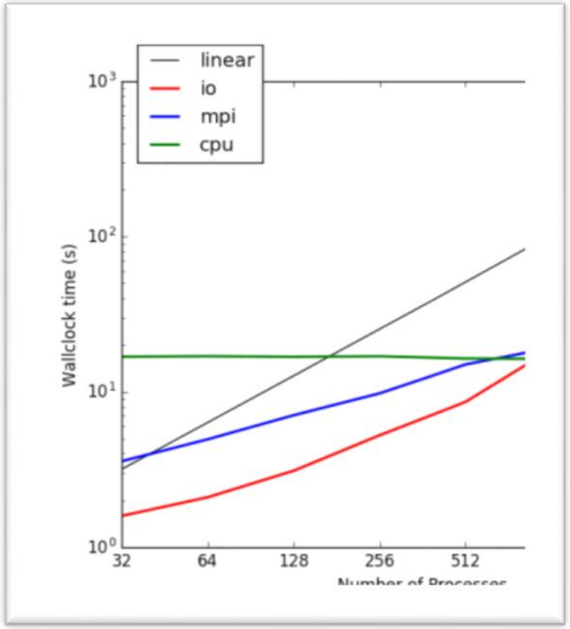
- Data export
  - JSON export
  - Data analytics (python)
  - Continuous integration



Scheduler statistics for whole system

Scheduler information collected from 'qstat' command

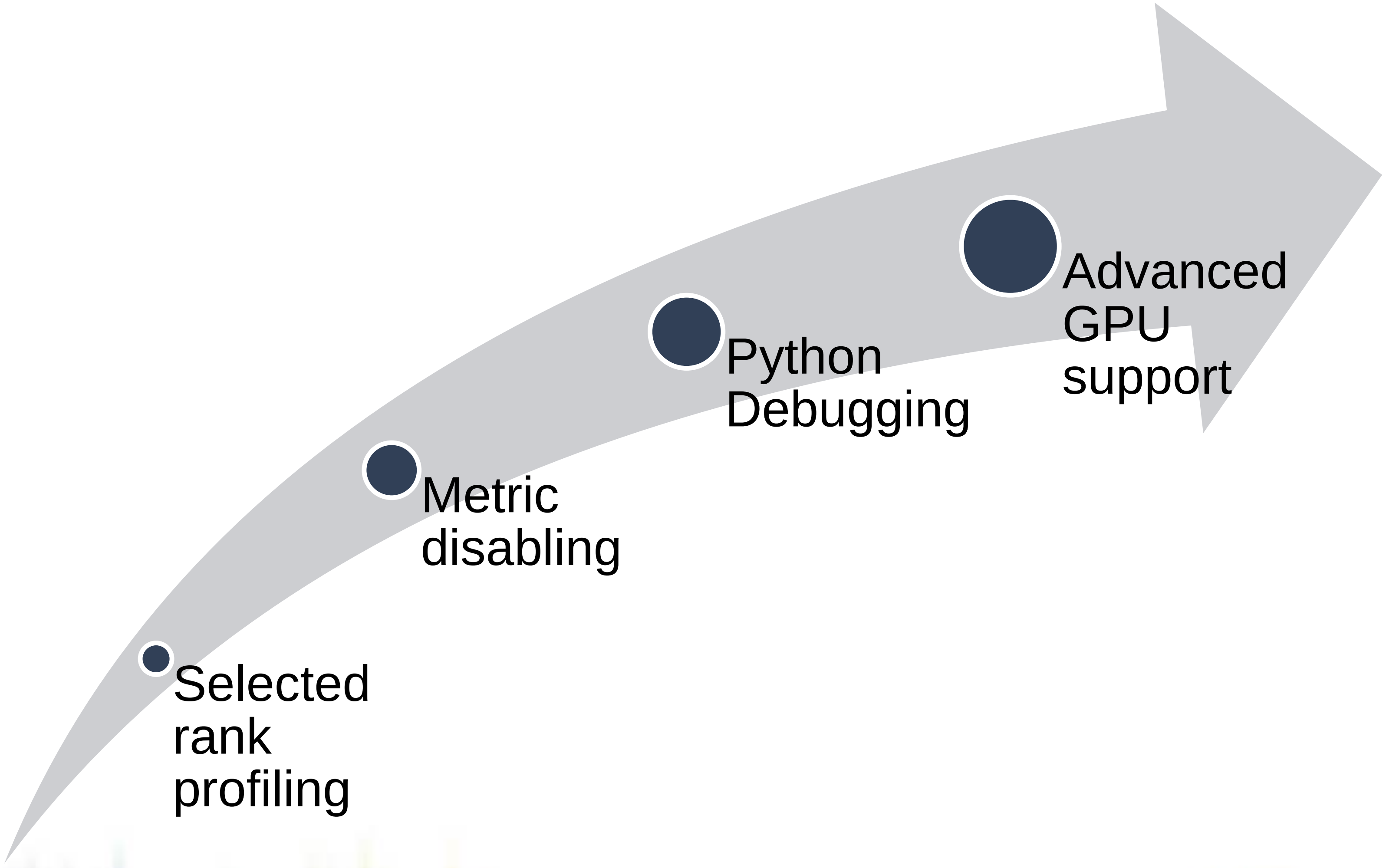
Maximum nodes in use	3.65 k	🟥
Maximum running jobs	315	🟥
Maximum queued jobs	226	🟥



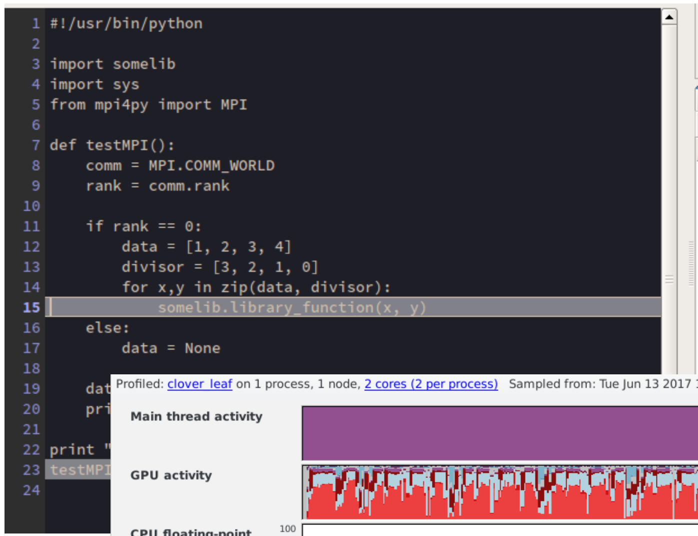
Building a Community

- Custom metrics can help others
  - Open source our metrics
  - Share them with different users
  - Build a centralised repository – and knowledge base
- Data processing
  - https://github.com/arm-hpc/allinea_json_analysis
  - Open source Python scripts for analysis
  - Strong and weak scaling analysis over multiple files
- Porting recipes
  - Share tools experience to aid porting

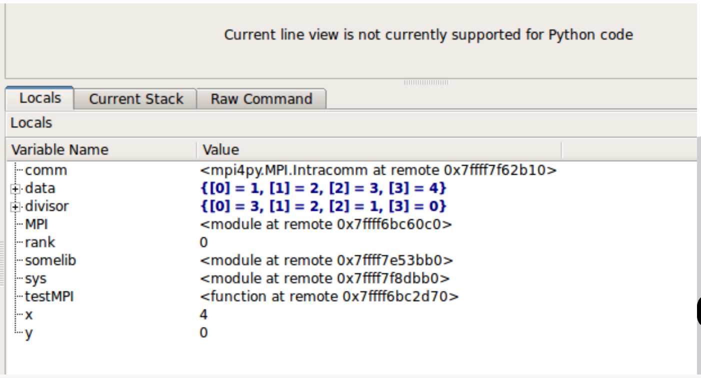
So Much More to Come



So Much More to Come



```
#!/usr/bin/python

import somelib
import sys
from mpi4py import MPI

def testMPI():
    comm = MPI.COMM_WORLD
    rank = comm.rank

    if rank == 0:
        data = [1, 2, 3, 4]
        divisor = [3, 2, 1, 0]
        for x,y in zip(data, divisor):
            somelib.library_function(x, y)
    else:
        data = None

    dat
    pri

print "
testMPI
```



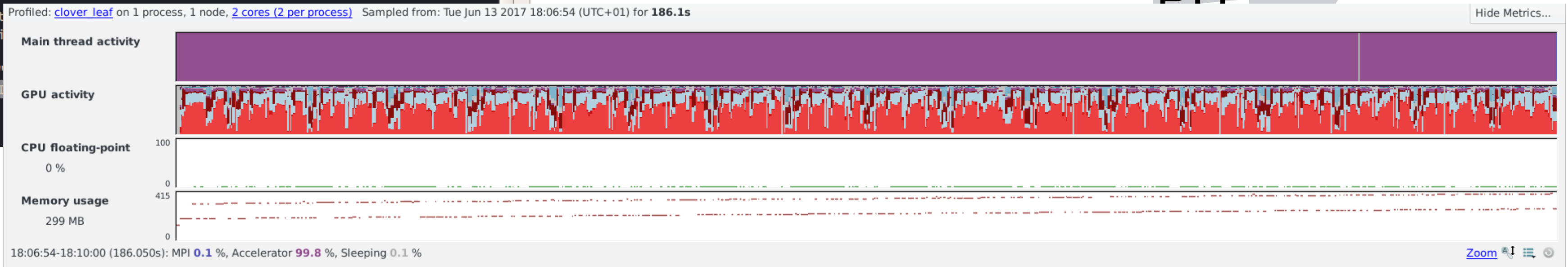
Current line view is not currently supported for Python code

Locals | Current Stack | Raw Command

Locals

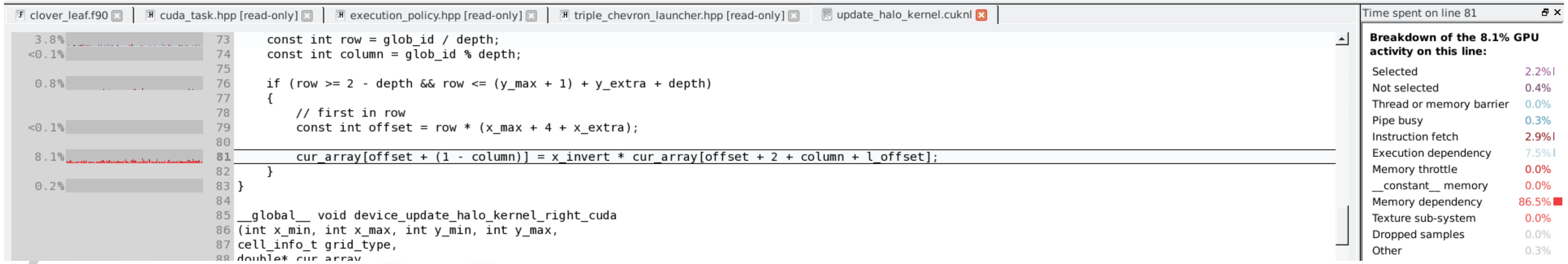
Variable Name	Value
comm	<mpi4py.MPI.Intracomm at remote 0x7ffff7f62b10>
data	{[0] = 1, [1] = 2, [2] = 3, [3] = 4}
divisor	{[0] = 3, [1] = 2, [2] = 1, [3] = 0}
MPI	<module at remote 0x7ffff6bc60c0>
rank	0
somelib	<module at remote 0x7ffff7e53bb0>
sys	<module at remote 0x7ffff7f8dbb0>
testMPI	<function at remote 0x7ffff6bc2d70>
x	4
y	0

dvanced



Profiled: clover_leaf on 1 process, 1 node, 2 cores (2 per process) Sampled from: Tue Jun 13 2017 18:06:54 (UTC+01) for **186.1s**

18:06:54-18:10:00 (186.050s): MPI 0.1 %, Accelerator 99.8 %, Sleeping 0.1 %



clover_leaf.f90 | cuda_task.hpp [read-only] | execution_policy.hpp [read-only] | triple_chevron_launcher.hpp [read-only] | update_halo_kernel.cuknl

```
 3.8%   73     const int row = glob_id / depth;
<0.1%   74     const int column = glob_id % depth;
        75
 0.8%   76     if (row >= 2 - depth && row <= (y_max + 1) + y_extra + depth)
        77     {
        78         // first in row
<0.1%   79         const int offset = row * (x_max + 4 + x_extra);
        80
 8.1%   81         cur_array[offset + (1 - column)] = x_invert * cur_array[offset + 2 + column + l_offset];
        82     }
 0.2%   83 }
        84
        85 __global__ void device_update_halo_kernel_right_cuda
        86 (int x_min, int x_max, int y_min, int y_max,
        87 cell_info_t grid_type,
        88 double* cur_array
```

Time spent on line 81

Breakdown of the 8.1% GPU activity on this line:

Selected	2.2%
Not selected	0.4%
Thread or memory barrier	0.0%
Pipe busy	0.3%
Instruction fetch	2.9%
Execution dependency	7.5%
Memory throttle	0.0%
__constant__ memory	0.0%
Memory dependency	86.5%
Texture sub-system	0.0%
Dropped samples	0.0%
Other	0.3%

Thank You

Oliver Perks
olly.perks@arm.com