Grm

Arm Forge Debugging and Optimization Tools for HPC

Beau Paisley<Beau.Paisley@arm.com> ATPESC2022

Arm Forge

An interoperable toolkit for debugging and profiling



Commercially supported by Arm





The de-facto standard for HPC development

- Most widely-used debugging and profiling suite in HPC
- Fully supported by Arm on Intel, AMD, Arm, IBM Power, Nvidia GPUs, etc.

State-of-the art debugging and profiling capabilities

- Powerful and in-depth error detection mechanisms (including memory debugging)
- Sampling-based profiler to identify and understand bottlenecks
- Available at any scale (from serial to petaflopic applications)

Easy to use by everyone

- Unique capabilities to simplify remote interactive sessions
- Innovative approach to present quintessential information to users

DDT Debugger Highlights









9 Step guide: optimizing high performance applications

arm

Improving the efficiency of your parallel software holds the key to solving more complex research problems faster. This pragmatic, 9 Step best practice guide will help you identify and focus on application readiness, bottlenecks and optimizations one step at a time.



Arm Performance Reports





Summary: clover_leaf is Compute-bound in this configuration

Compute	100.0%	Time spent running application code. High values are usually good. This is very high ; check the CPU performance section for advice
MPI	0.0%	Time spent in MPI calls. High values are usually bad. This is very low ; this code may benefit from a higher process count
I/O	0.0%	Time spent in filesystem I/O. High values are usually bad. This is negligible ; there's no need to investigate I/O performance

This application run was Compute-bound. A breakdown of this time and advice for investigating further is in the CPU Metrics section below.

As very little time is spent in MPI calls, this code may also benefit from running at larger scales.

MPI

A breakdown of the 0.0% MPI time:			
Time in collective calls	0.0%		
Time in point-to-point calls	0.0%		
Effective process collective rate	0.00 bytes/s		
Effective process point-to-point rate	0.00 bytes/s		

No time is spent in MPI operations. There's nothing to optimize here!

OpenMP

А	breakdown	of	the	99	7%	time ir	۱ <mark>O</mark> p	enMP	regions:	
---	-----------	----	-----	----	----	---------	--------------------	------	----------	--

Computation	85.6%	
Synchronization	14.4%	1 - C
Physical core utilization	8.3%	1
System load	7.8%	1

Physical core utilization is low and some cores may be unused. Try increasing OMP_NUM_THREADS to improve performance.

1.	\sim	
1)	U	

A breakdown of the 0.0% I/	0 time:	
Time in reads	0.0%	T.
Time in writes	0.0%	T.
Effective process read rate	0.00 bytes/s	T.
Effective process write rate	0.00 bytes/s	I

No time is spent in I/O operations. There's nothing to optimize here!

Memory

Per-process memory usage may also affect scaling:				
Mean process memory usage	312 MiB			
Peak process memory usage	314 MiB			
Peak node memory usage	2.0%			

The peak node memory usage is very low. Larger problem sets can be run before scaling to multiple nodes.



No source code needed

Explicit and usable output

MAP Source Code Profiler Highlights



arm

Python Profiling

- 19.0 adds support for Python
 - Call stacks
 - Time in interpreter
- Works with MPI4PY
 - Usual MAP metrics
- Source code view
 - Mixed language support

Note: Green as operation is on numpy array, so backed by C routine, not Python (which would be pink)



Forge Follow Up Materials

ANL specific references <u>https://www.alcf.anl.gov/support-center/theta/arm-ddt-theta</u> <u>https://www.alcf.anl.gov/support-center/theta/arm-map</u>

Getting started videos, https://developer.arm.com/tools-and-software/server-and-hpc/arm-architecture-tools/arm-forge/videos

Offline debugging blogs, https://community.arm.com/developer/tools-software/hpc/b/hpc-blog/posts/debugging-while-you-sleep https://community.arm.com/developer/tools-software/hpc/b/hpc-blog/posts/more-debugging-while-you-sleep-with-ddt

Topic specific Arm HPC webinars,

https://developer.arm.com/tools-and-software/server-and-hpc/arm-architecture-tools/training/arm-hpc-tools-webinars

Python specific references https://developer.arm.com/documentation/101136/2102/DDT/Get-started-with-DDT/Python-debugging https://developer.arm.com/documentation/101136/2102/MAP/Python-profiling

Arm Forge Overview Recorded for the SC Student Cluster Competition https://www.youtube.com/watch?v=Pe2WDJR2cTg&t=13s

Debugging methodology presentation at Nvidia GTC https://www.nvidia.com/en-us/on-demand/session/gtcspring22-s41737/