ARM HPC Ecosystem 2016

ARM

Chris Goodyer and Eric Van Hensbergen

SC16 – Salt Lake City, UT
November 14, 2016

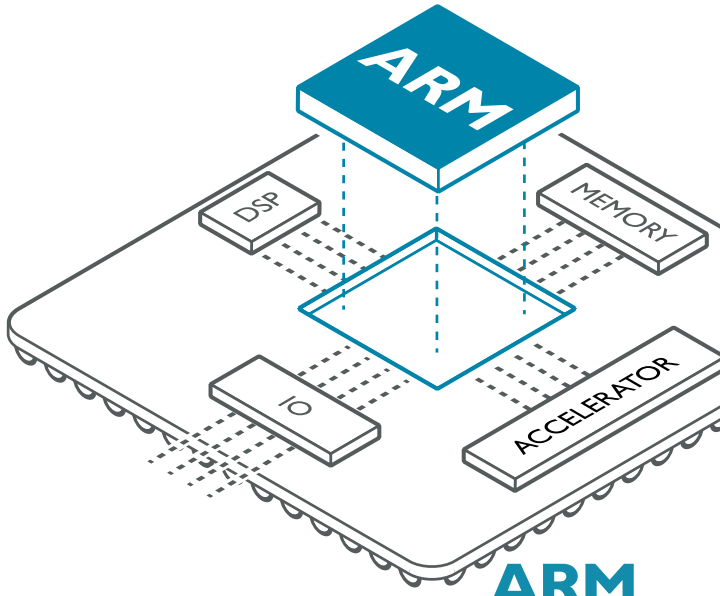
ARM in HPC

Energy efficiency
- Can target highly parallel applications with dense, high-core count SoCs
- Same ISA heterogeneity enabling "right-sized" cores

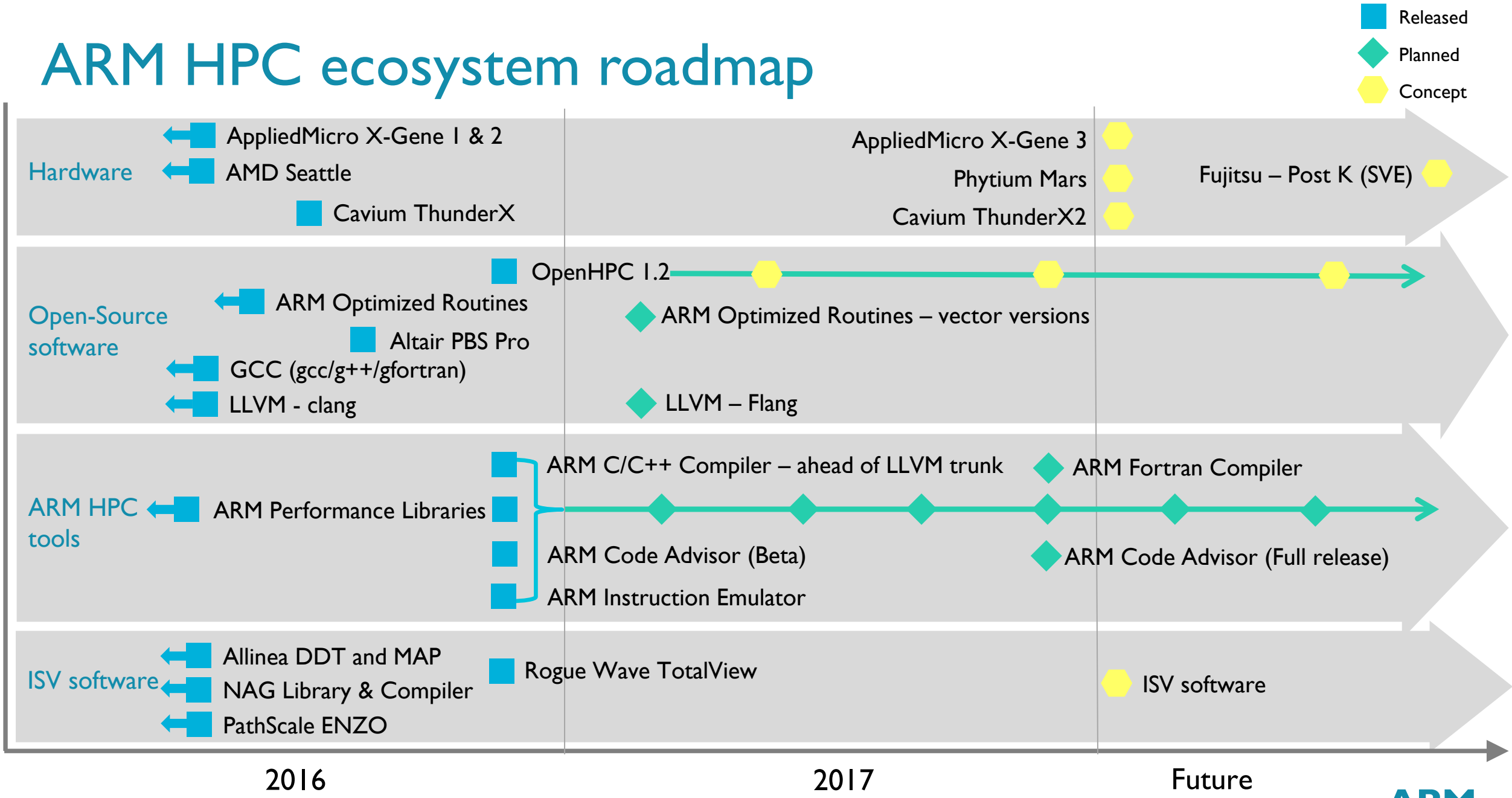
Ecosystem
- Don't reinvent the wheel
- Focus on differentiation

End-user choice
- Independent silicon providers within a shared software ecosystem
- Wide range of price/performance/power/size options available

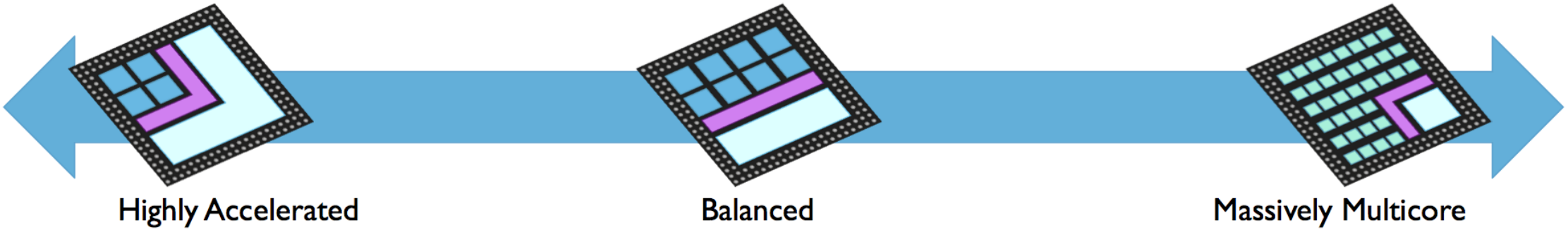
Customization
- ARM's partners can build SoCs very quickly
- Extensions like SVE allow for differentiation
- Low-cost SoC development interesting in the US:
  - SoC for HPC: http://www.socforhpc.org
  - DARPA CRAFT: http://www.darpa.mil/news-events/2015-08-17



ARM HPC ecosystem roadmap



Range of SoCs addressing infrastructure



XILINX
MPSoC

freescale
QorIQ LS2

applied micro
X-Gene™

EZCHIP TECHNOLOGIES
Tile-MX 100

ALTERA
Stratix® 10

AMD
Opeteron™ A1100

CAVIUM
ThunderX

One Size Does Not Fit All

ARM

Open source in the ARM HPC ecosystem

- Over the past twelve months many more packages and applications have been successfully ported to ARM HPC

OpenFOAM

NAMD
Scalable Molecular Dynamics

GROMACS FAST. FLEXIBLE. FREE.

QUANTUMESPRESSO

Geant 4

LAMMPS

WRF

OpenHPC – now on ARM

OpenHPC is a community effort to provide a common, verified set of open source packages for HPC deployments

ARM's participation:

- Silver member of OpenHPC
- ARM is on OpenHPC Technical Steering Committee in order to drive ARM architecture build support

Status (November 2016):

- 1.2.0 release out now
- All packages built on ARMv8 for both CentOS and SUSE
- ARM-based machines are being used for building and also in the OpenHPC build infrastructure

Functional Areas	Components include
Base OS	RHEL/CentOS 7.1, SLES 12
Administrative Tools	Conman, Ganglia, Lmod, LosF, ORCM, Nagios, pdsh, prun
Provisioning	Warewulf
Resource Mgmt.	SLURM, Munge. Altair PBS Pro*
I/O Services	Lustre client (community version)
Numerical/Scientific Libraries	Boost, GSL, FFTW, Metis, PETSc, Trilinos, Hypre, SuperLU, Mumps
I/O Libraries	HDF5 (pHDF5), NetCDF (including C++ and Fortran interfaces), Adios
Compiler Families	GNU (gcc, g++, gfortran)
MPI Families	OpenMPI, MVAPICH2
Development Tools	Autotools (autoconf, automake, libtool), Valgrind,R, SciPy/NumPy
Performance Tools	PAPI, Intel IMB, mpiP, pdtoolkit TAU

Open source compiler highlights in 2016

- Performance Improvements:
  - GCC on ARM Cortex-A57 processor: 12% SPECCPU improvement May 2013 to Nov 2016 (GCC 4.8 to GCC 6)
    - drop in rate of improvement is expected in GCC because of increasing maturity of the compiler.
  - LLVM currently tracks around 3% behind GCC
  - LLVM code size optimization
- Architecture enablement on both GCC and LLVM:
  - ARMv8.1-A & ARMv8.2-A, ARMv8.3-A , ARMv8-M, ARMv8-R
- ARM was in the top five contributors to GCC 6
  - AArch64 and ARM are now both primary platforms of GCC
- Work in progress
  - Vector-math library for ARM/AArch64: enabling more vectorization opportunities
  - GCC store merging: a target independent pass to combine stores
  - GCC shrink wrapping AArch64: enable general shrink wrap enhancement on AArch64

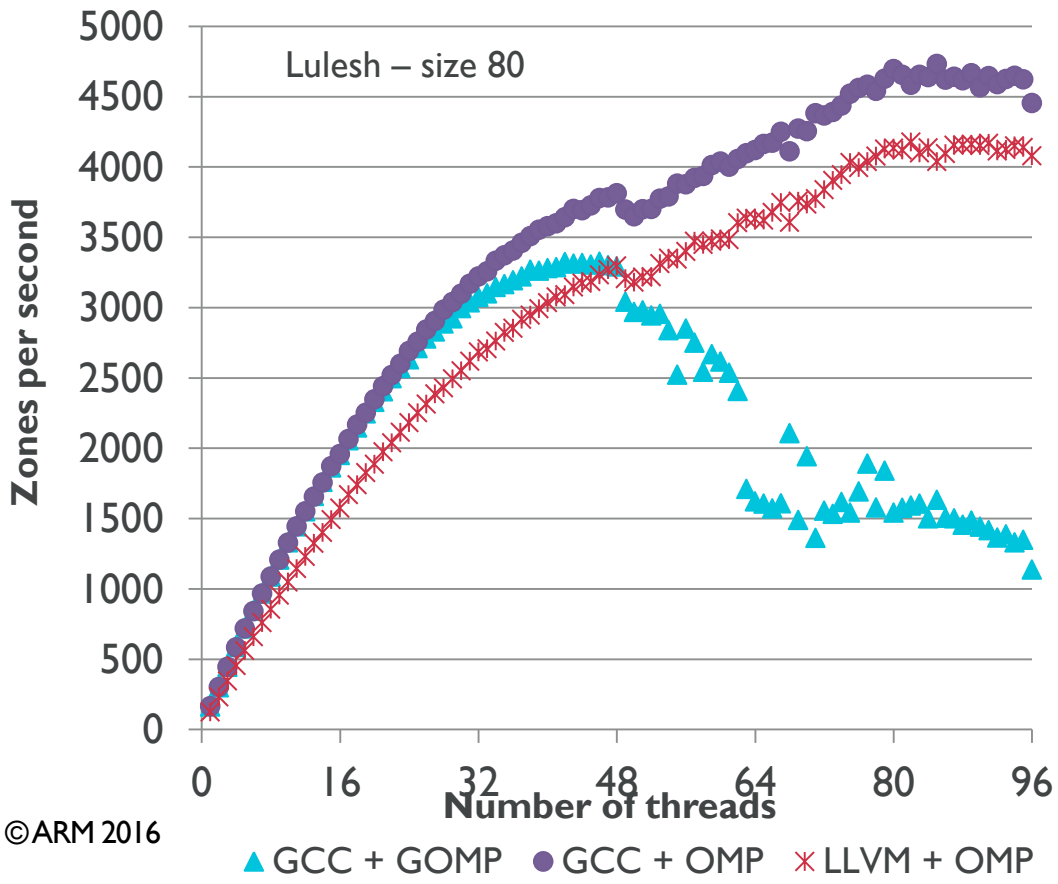
Fortran support in LLVM

- Last November the US Department of Energy funded PGI to open source their compiler front-end for LLVM
  - We are reviewing the source code and recommending enhancements
  - We have demonstrated that this compiler can generate correct AArch64 code

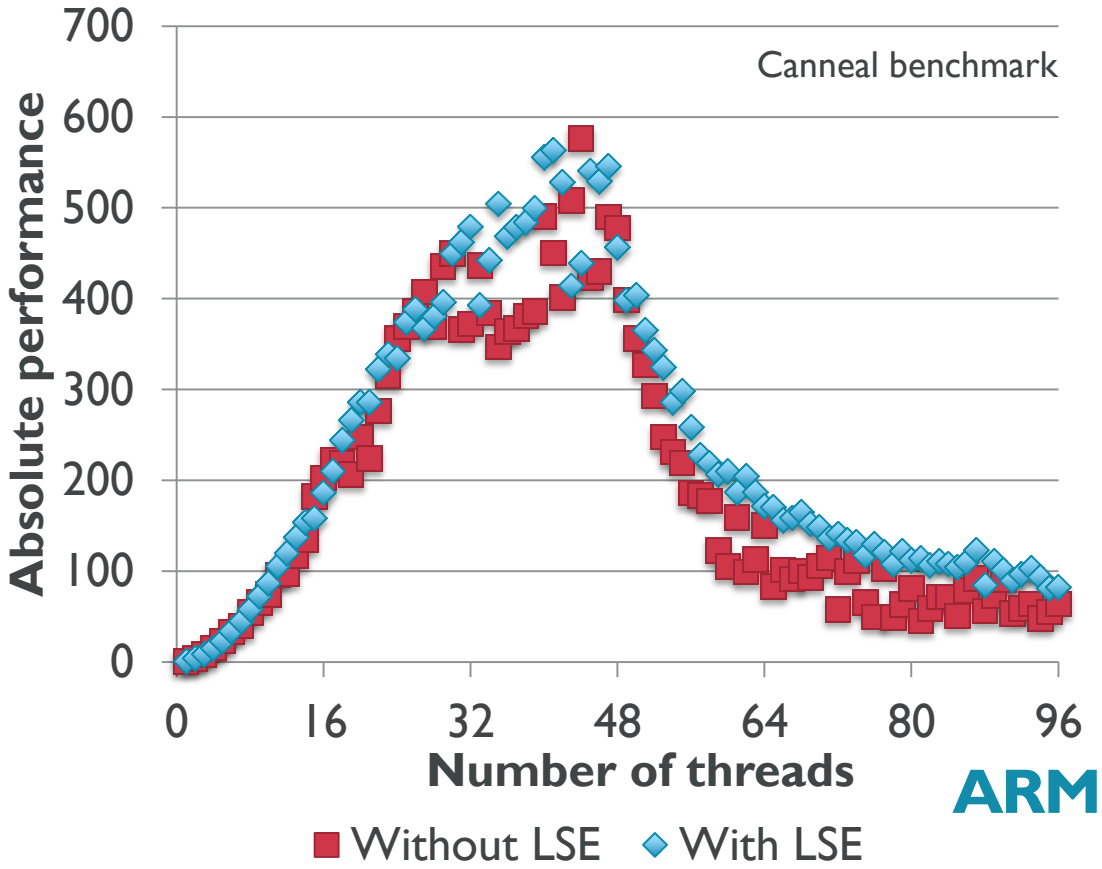
- ARM is developing vector math functions for Fortran runtime library
  - Apache 2 licensed as part of the ARM Optimized Routines repository on github
    - https://github.com/ARM-software/optimized-routines
  - Will be used by the PGI Flang compiler
  - Will be upstreamed to glibc and LLVM parallel-libs

LLVM OpenMP development

- We have been contributing ARM-related upstream patches to LLVM's 'libomp'
- libomp demonstrates superior scaling, even when used with GCC



- Including LSE (ARMv8.1-A) instructions in both GCC and libomp give enhanced performance at large core-counts



ISV slides

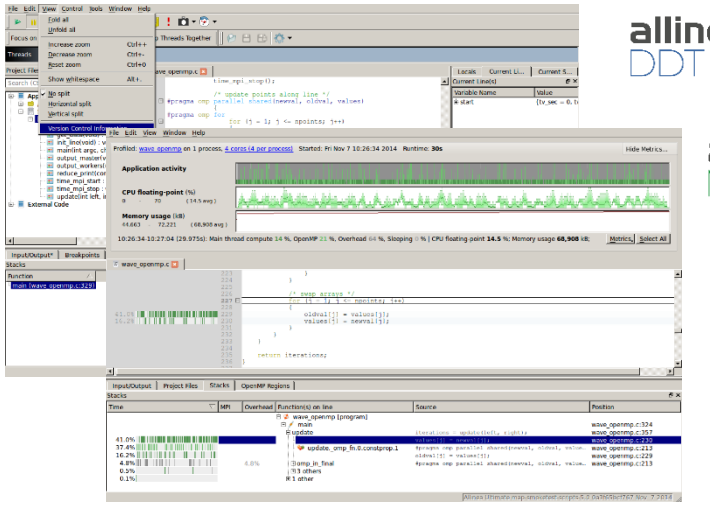
Allinea's tools for HPC

Allinea Forge

Debug and profile codes

Performance Reports

Monitor and tune applications



allinea DDT

allinea MAP

CPU

A breakdown of the 88.5% CPU time:

Single-core code	100.0%
Scalar numeric ops	22.4%
Vector numeric ops	0.0%
Memory accesses	77.6%

The per-core performance is memory-bound. Use a profiler to identify time-consuming loops and check their cache performance.

No time is spent in vectorized instructions. Check the compiler's vectorization advice to see why key loops could not be vectorized.

MPI

A breakdown of the 11.4% MPI time:

Time in collective calls	3.1%
Time in point-to-point calls	96.9%
Effective process collective rate	31.7 kB/s
Effective process point-to-point rate	269 kB/s

Most of the time is spent in point-to-point calls with a very low transfer rate. This suggests load imbalance is causing synchronization overhead; use an MPI profiler to investigate further.

I/O

A breakdown of the 0.0% I/O time:

Time in reads	0.0%
Time in writes	0.0%
Effective process read rate	0.00 bytes/s
Effective process write rate	0.00 bytes/s

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

Threads

A breakdown of how multiple threads were used:

Computation	0.0%
Synchronization	0.0%
Physical core utilization	100.0%
Involuntary context switches per second	1.8

No measurable time is spent in multithreaded code.

Memory

Per-process memory usage may also affect scaling:

Energy

A breakdown of how the total 588 J energy was spent:

Already supported for ARMv8

NAG and ARM

NAG's experts help create ARM Performance Libraries

- Updates continue to be shipped (inc. LAPACK 3.6.1)

Product availability for 64-bit ARMv8-A

- NAG Library
  - largest commercially available collection of numerical and statistical algorithms (> 1800 functions)
  - *Scales well, takes advantage of ARM Performance Libraries*
  - Tested on Juno (ARM) and ThunderX (Cavium)
- NAG Fortran Compiler
  - Fortran 95, 2003 & 2008 and OpenMP support,
  - World's best checking compiler

https://www.nag.com/nag-partner-arm

support@nag.co.uk

ARM HPC tools portfolio

ARM C/C++ Compiler

COMMERCIALLY SUPPORTED
FOR HPC APPLICATIONS

ARM Performance Libraries

BLAS, LAPACK and FFT
MICRO-ARCHITECTURALLY TUNED

ARM Code Advisor

ACTIONABLE ADVICE TO
OPTIMIZE YOUR CODE

ARM SVE C/C++ Compiler

COMPILER SUPPORT FOR
ARM SCALABLE VECTOR EXTENSION

ARM Instruction Emulator

DEVELOP SOFTWARE FOR
TOMORROW'S HARDWARE TODAY

HPC leadership: international exascale programs

- **United States**
  ARM is currently a participant in two Department of Energy funded Exascale projects: Data Movement Dominates and Fast Forward. ARM has also recently submitted proposals for involvement in the Path Forward program which includes $300M funding to develop Exascale technologies.

- **European Union**
  Through FP7 and Horizon 2020, ARM has been involved in several funded pre-Exascale projects including the Mont Blanc program which deployed one of the first ARM prototype HPC systems.

- **Japan**
  At ISC2016, Fujitsu and RIKEN announced that the Post-K system targeted at Exascale will be based on ARMv8 and custom extensions which were described at HotChips 2016.

- **China**
  James Lin, vice director for the Center of HPC at Shanghai Jiao Tong University claims China will build three pre-Exascale prototypes to select the architecture for their Exascale system. The three prototypes are based on AMD, SunWei TaihuLight, and ARMv8-A.

ARM

http://www.arm.com/hpc

Contact: hpc@arm.com