Andrei Poenaru

University of Bristol /

GW4 Alliance

Early toolchain performance experiments

On Isambard, the world's first production 64-bit Arm supercomputer

University of BRISTOL

EPSRC

GW4

'Isambard' is a new UK Tier 2 HPC service from GW4

University of BRISTOL

UNIVERSITY OF BATH

UNIVERSITY OF EXETER

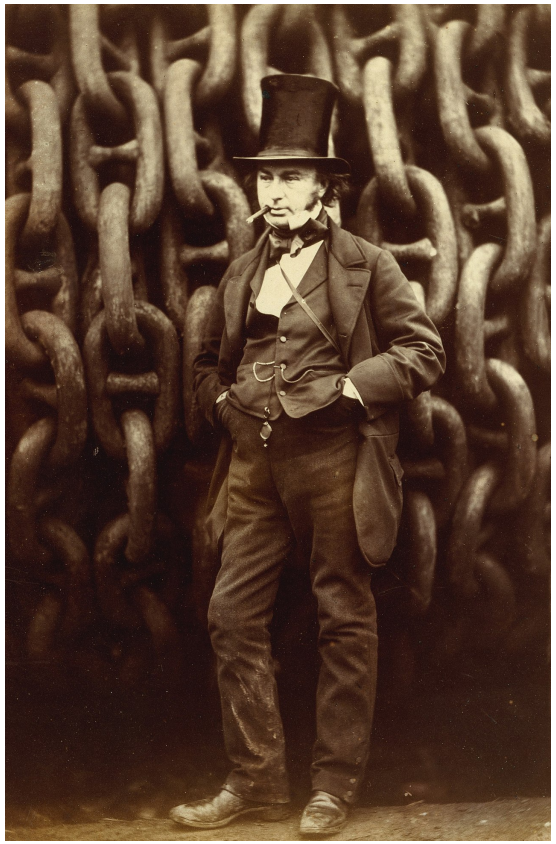
CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

Met Office

EPSRC

CRAY
THE SUPERCOMPUTER COMPANY

ARM



Isambard Kingdom Brunel
1804-1859

University of BRISTOL

EPSRC

GW4

Isambard system specifications

- **10,752** ARMv8 cores (168 x 2 x 32)
  - **Cavium ThunderX2, 32 cores, 2.1 GHz**
- Cray XC50 Scout form factor
- High-speed **Aries** interconnect
- Cray HPC optimised software stack
  - CCE, CrayPAT, Cray MPI, math libraries, …
- **Technology comparison:**
  - **x86, Xeon Phi, Pascal GPUs**
- Phase 1 installed March 2017
- Phase 2 (the Arm part) currently in bring-up
- £4.7m total project cost over 3 years

University of BRISTOL

EPSRC

Isambard's core mission: deploying Arm in production HPC

Starting by optimizing codes from the top 10 most heavily used on ARCHER:

- **VASP, CASTEP, GROMACS, CP2K, UM, NAMD, Oasis, SBLI, NEMO**
- Most of these codes are written in FORTRAN

Additional important codes for project partners:

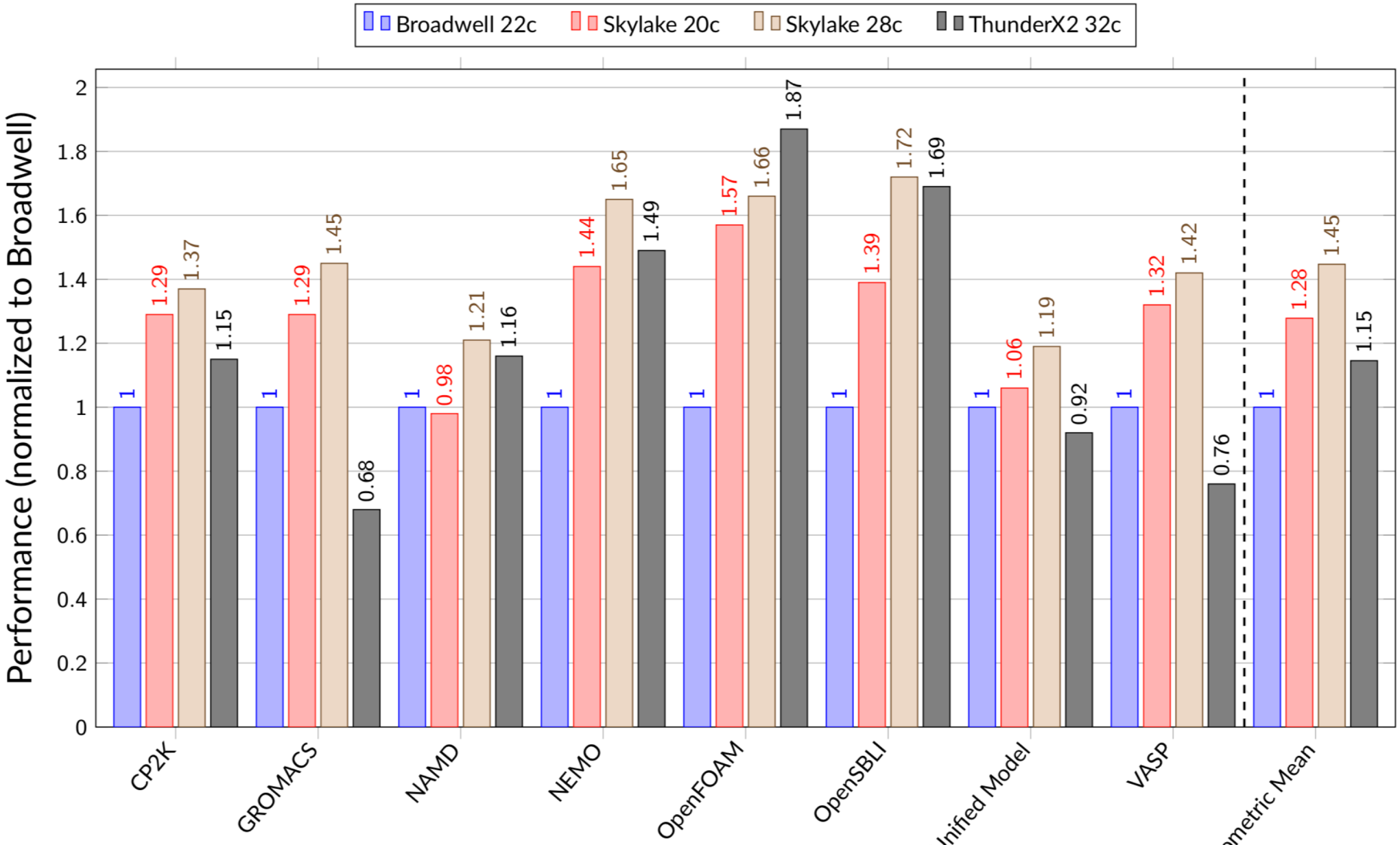
- **OpenFOAM, OpenIFS, WRF, CASINO, LAMMPS, …**

Getting started with running on Arm

- Our experience has been very smooth: virtually no porting required, most things work **out-of-the-box**!

- No unusual compiler flags needed:
  - Arm: `-mcpu=thunderx2t99 -O3 -ffast-math`
  - GCC: `-march=armv8.1-a -mcpu=thunderx2t99 -O3 -ffast-math -ffp-contract=fast`
  - CCE: `module load craype-arm-thunderx2`

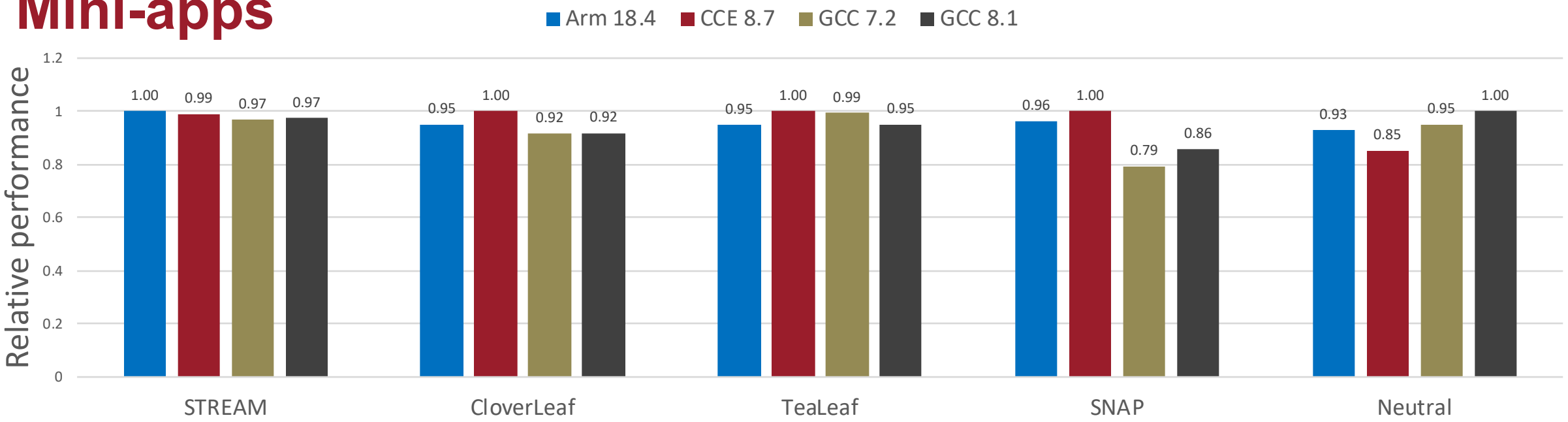
Performance on heavily used applications from ARCHER



More Isambard details and performance results on Wednesday…

Benchmark	ThunderX2	Broadwell	Skylake
STREAM	Arm 18.3	Intel 18	CCE 8.7
CloverLeaf	CCE 8.7	Intel 18	Intel 18
TeaLeaf	CCE 8.7	GCC 7	Intel 18
SNAP	CCE 8.6	Intel 18	Intel 18
Neutral	GCC 8	Intel 18	GCC 7
CP2K	GCC 8	GCC 7	GCC 7
GROMACS	GCC 8	GCC 7	GCC 7
NAMD	Arm 18.2	GCC 7	GCC 7
NEMO	CCE 8.7	CCE 8.7	CCE 8.7
OpenFOAM	GCC 7	GCC 7	GCC 7
OpenSBLI	CCE 8.7	Intel 18	CCE 8.7
UM	CCE 8.6	CCE 8.5	CCE 8.7
VASP	GCC 7.2	Intel 18	Intel 18

Mini-apps



- For each application, results are relative to best configuration
- CCE/Arm fastest in all but one case
  - Biggest factor is vectorisation
  - Neutral does not vectorise at all; GCC 8 also produces best code on BDW and SKL
- Going from GCC 7.2 to 8.1 loses performance in TeaLeaf

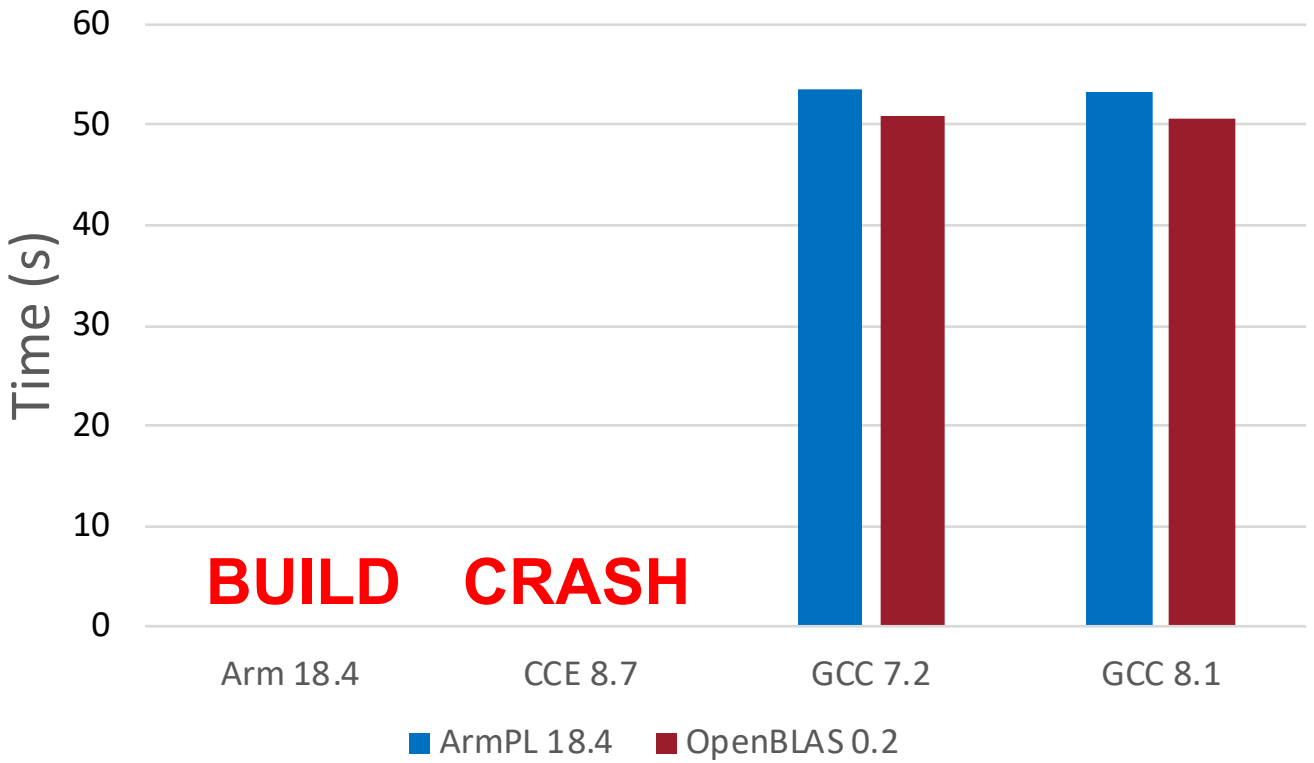
University of BRISTOL

EPSRC

GW4

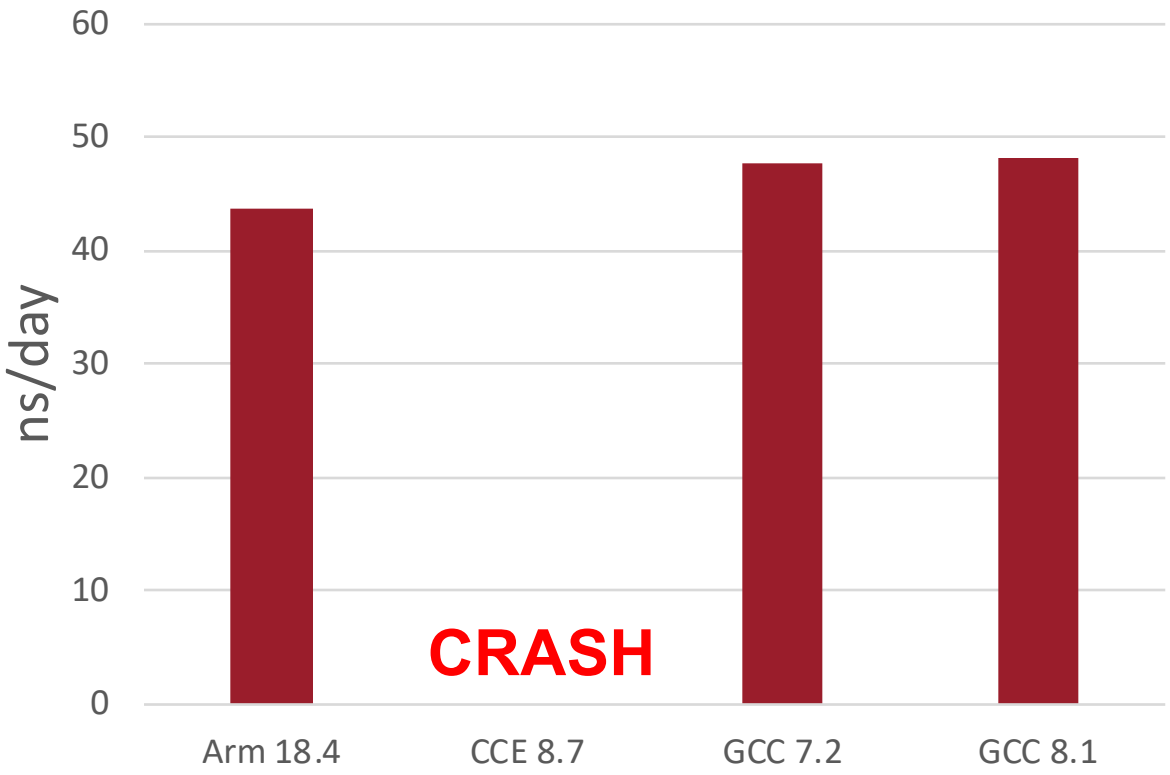
CP2K

- Benchmark: H2O-64
- Uses BLAS and FFTs
- ArmPL cannot be used for FFTs (unimplemented interface)
- CCE: segmentation fault at run time
  - This happens on Arm and SKL, but not BDW



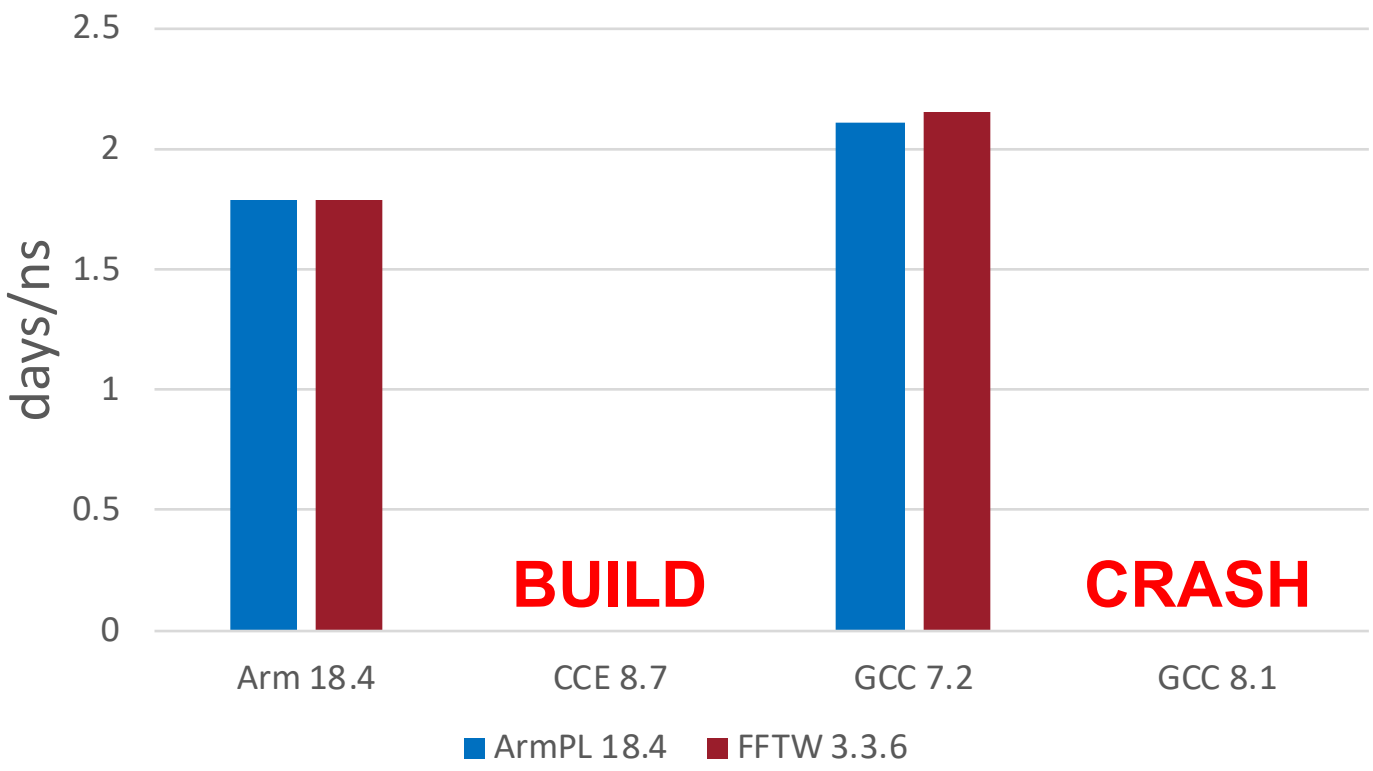
GROMACS

- Benchmark: ion channel, uses PME
  - Higher is better here!
- Hand-optimised NEON, ~35% of time in FFTs
- Build failure ArmPL (unimplemented interface)
  - With small code change, performance is similar to FFTW
- Run-time crash with CCE
  - Same on x86



NAMD

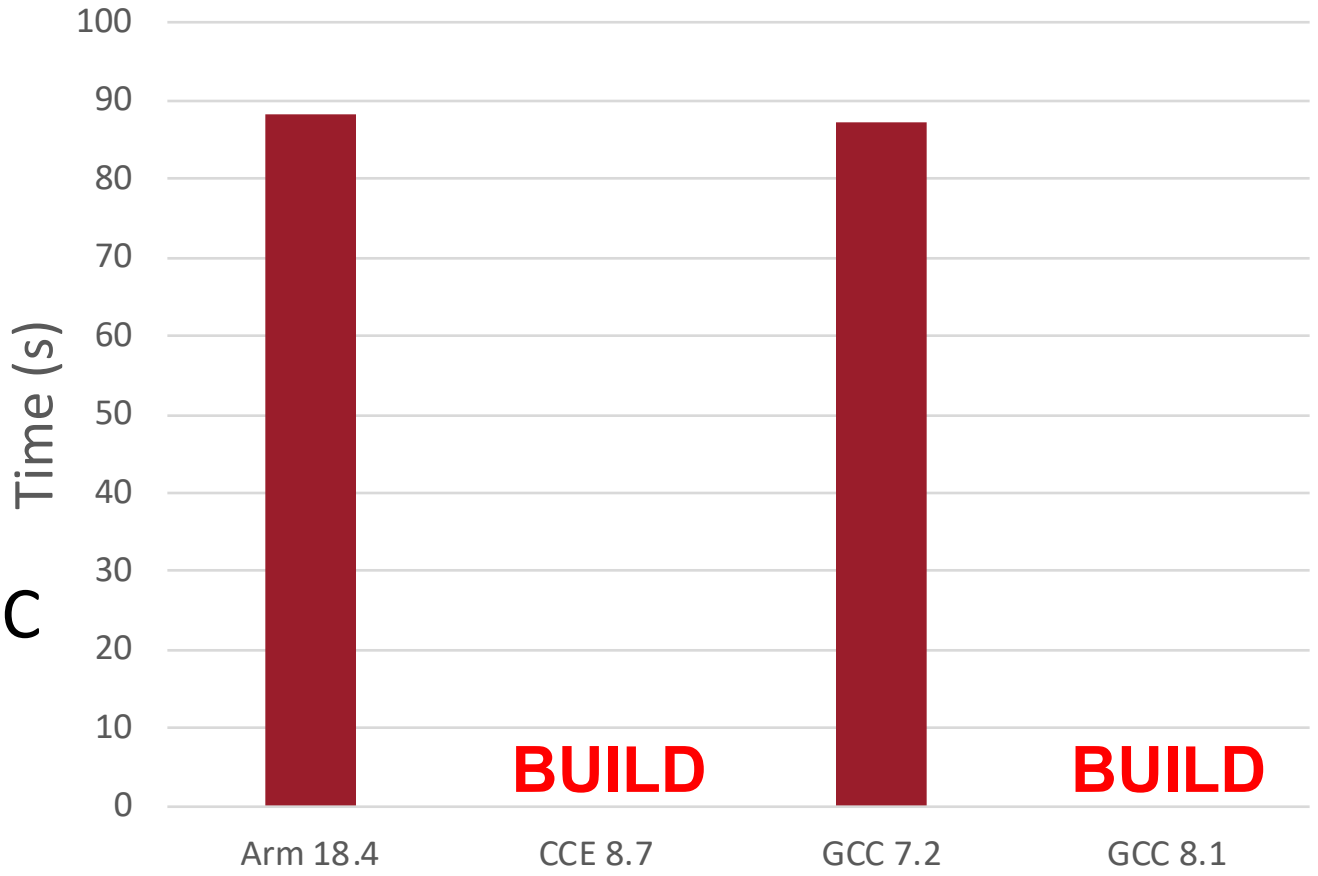
- Benchmark: STMV, uses PME
- FFTs only ~2% of run time
- No MPI, Charm++ threading
- GCC 8 hangs at run time
  - May be a Charm++ issue?
- CCE build also fails on x86



Library	Planning (s)	Execution (s)
ArmPL	0.12	162.04
FFTW	0.42	267.39

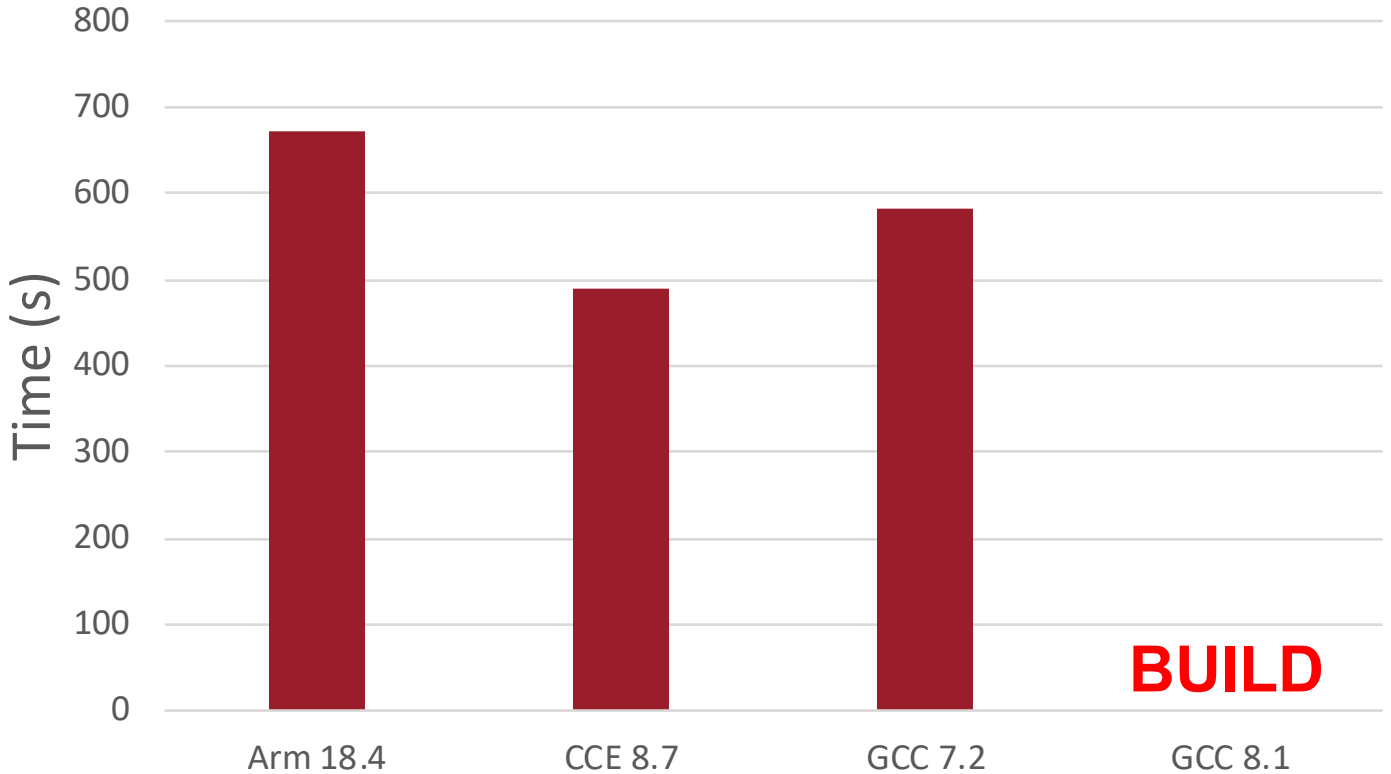
OpenFOAM

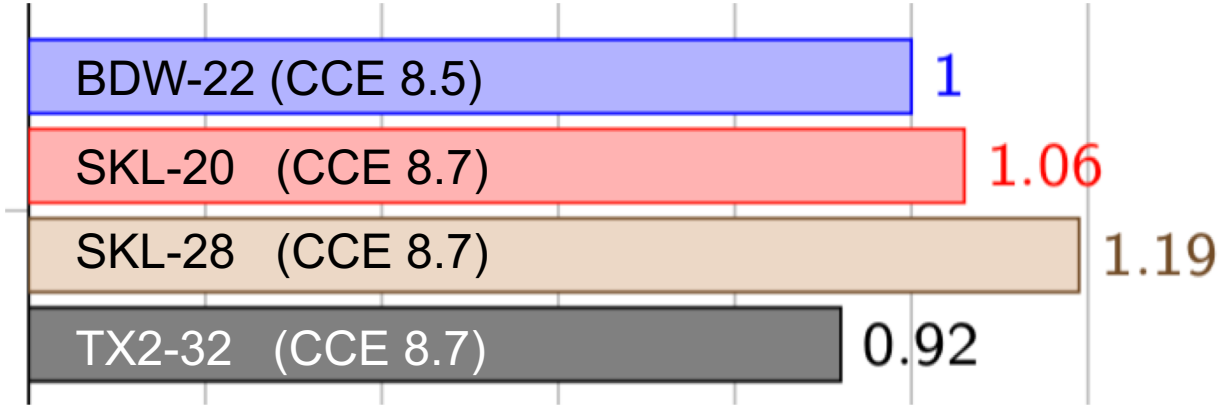
- Benchmark: RANS DrivAer case, `simpleFoam` solver
- Almost entirely bandwidth-bound, follows STREAM profile
- No maths library calls
- ~10% faster with SMT
  - 2-way with Arm, 4-way with GCC
- Build error with GCC 8, CCE
  - Not Arm-specific



Unified Model

- Benchmark: AMIP v10.8, provided by Met Office
- Bandwidth-bound, but single node not representative
  - Expect TX2 to scale better at 10s–100s nodes
- ICE on GCC 8
- On Broadwell, CCE 8.5 was fastest, but this is not available on Arm





University of BRISTOL

EPSRC

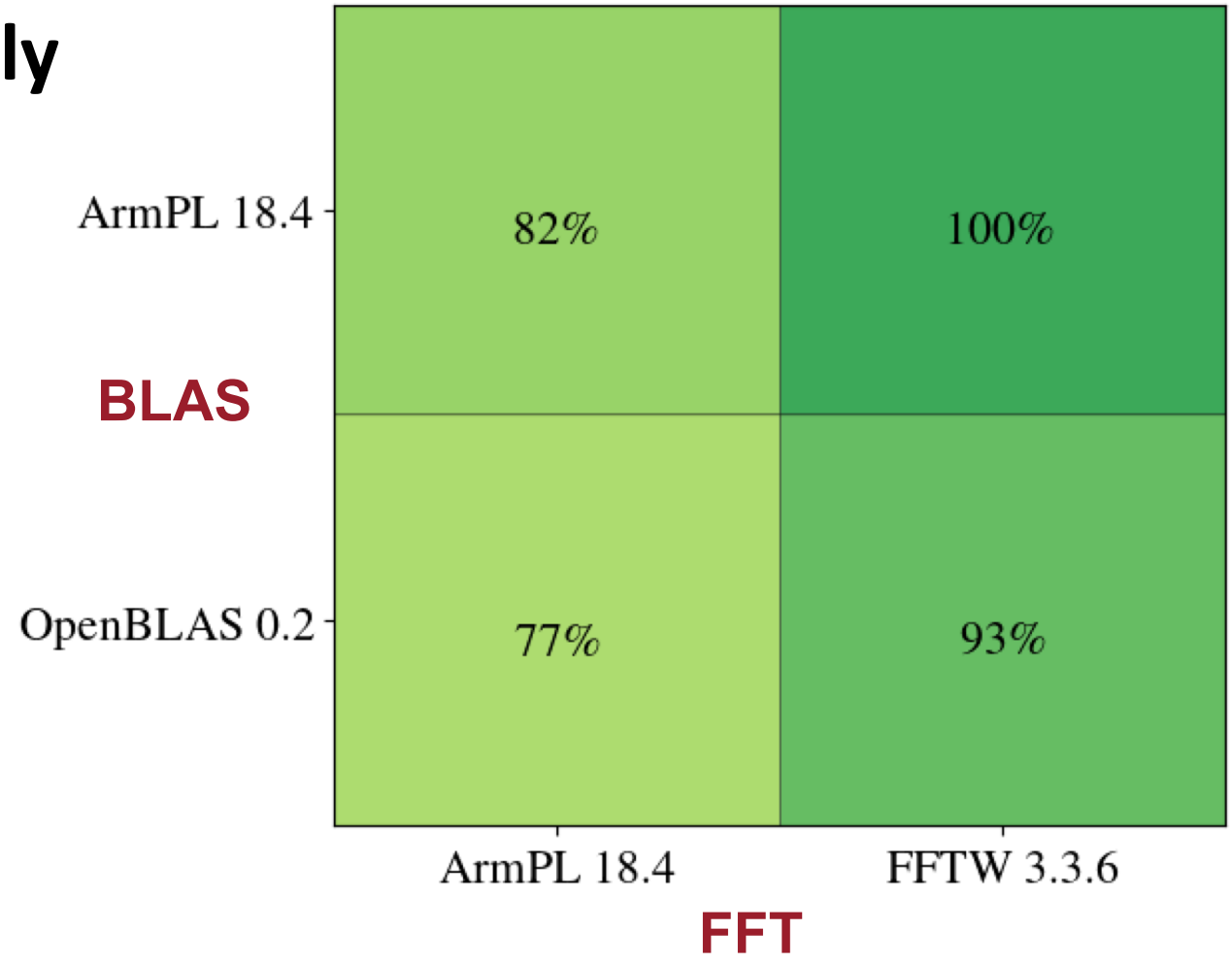
GW4

VASP

- These results are with GCC 7 **only**
  - Build errors with Arm, CCE, and GCC 8

- Cumulative FFT times
  - 64 MPI, 1 OMP

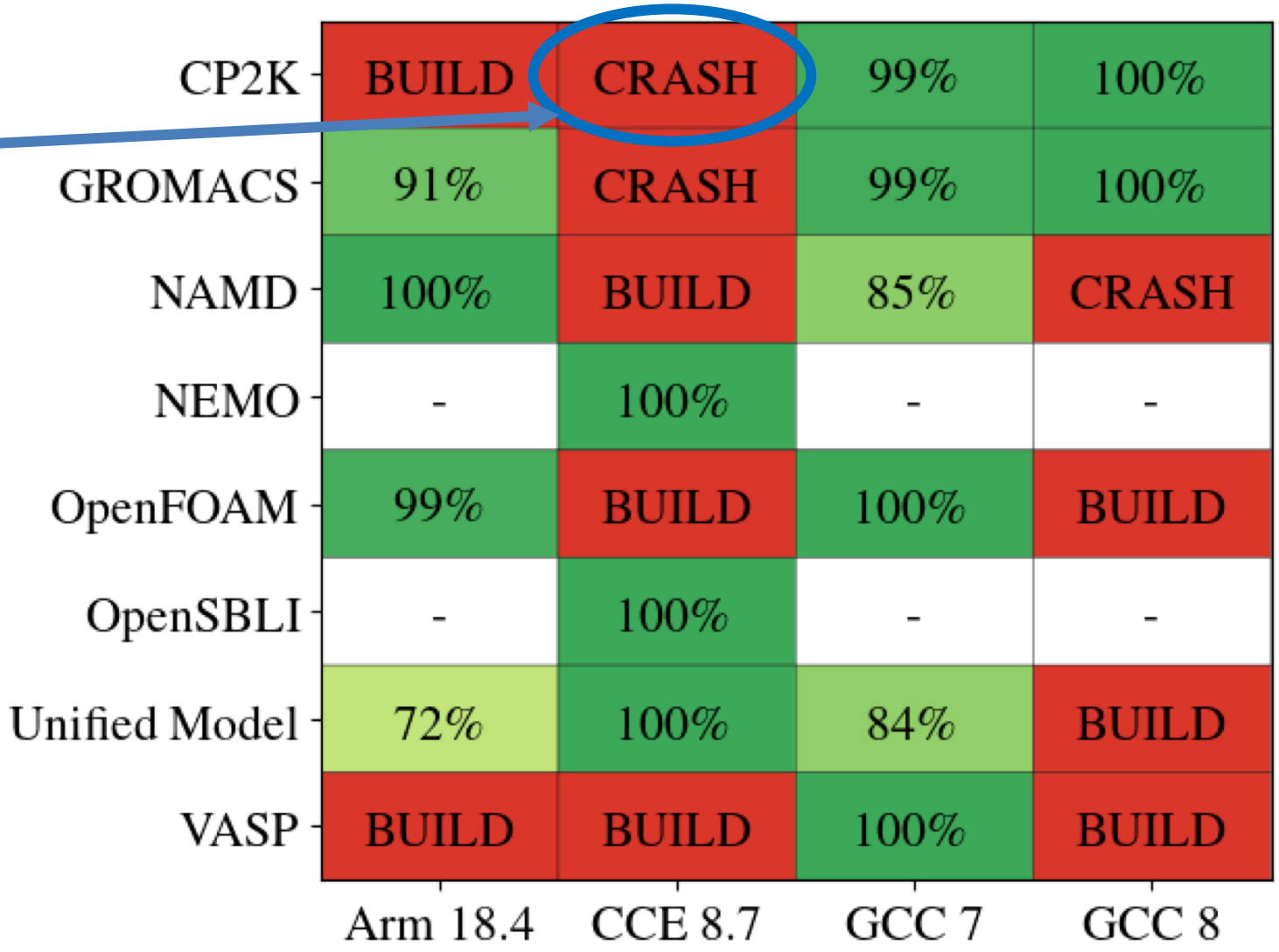
Library	Planning (s)	Execution (s)
ArmPL	874.4	2639.3
FFTW	308.5	3182.1



Compiler comparison

Only on Arm and SKL

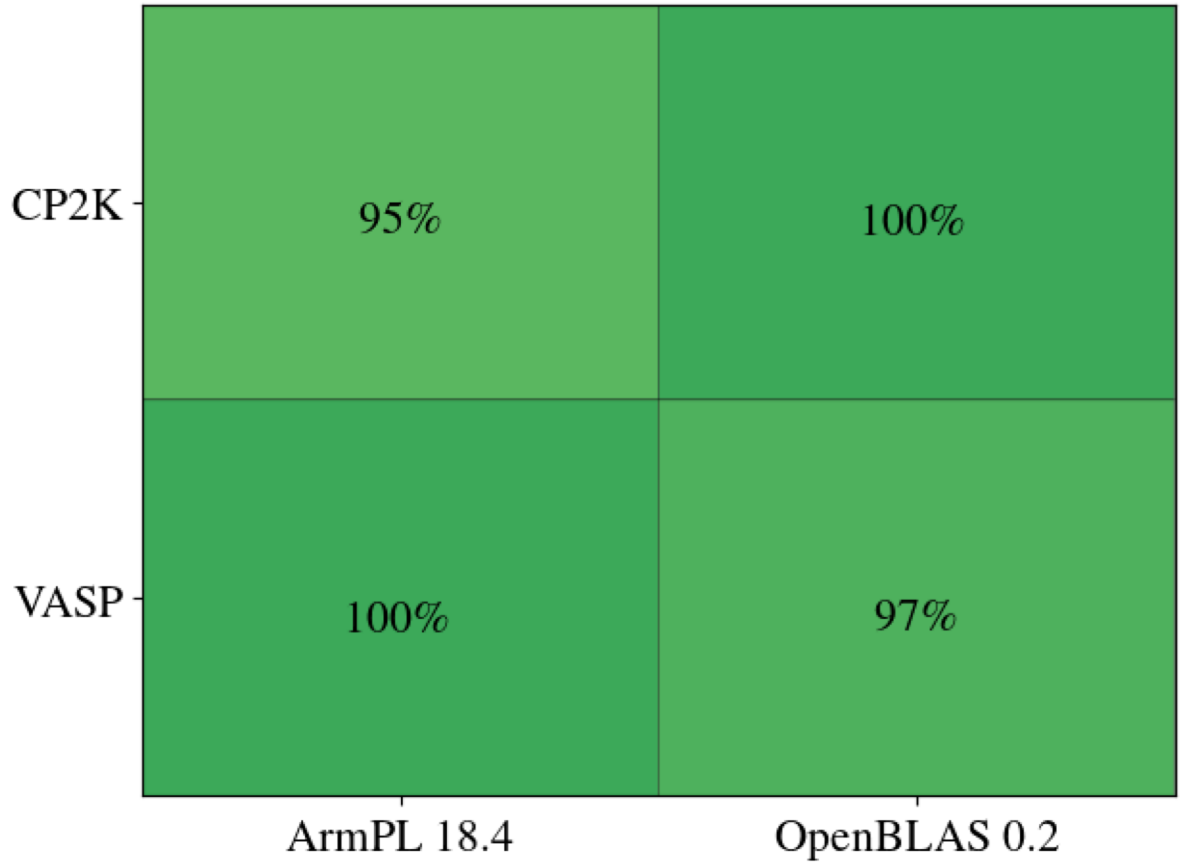
- Best libraries used in each case
- All CCE and GCC **issues present on x86 as well**
  - Only CP2K/CCE works on BDW
- NEMO and OpenSBLI still in progress…



Maths library comparison

FFT

BLAS



University of BRISTOL

EPSRC

GW4

Reproducibility

- We are maintaining a repository of **benchmarking scripts**: https://github.com/UoB-HPC/benchmarks
- All the (mini-)applications presented today are covered
  - Simple, single commands to build and run each benchmark
- Currently covered: TX2, BDW, SKL
  - You will likely need to adapt modules and paths for your own systems!
- All benchmarks include **output files**
- We will keep updating the repo as software and hardware evolves!

Conclusions

- Results show Arm performance is competitive with current high-end HPC CPUs, making a real alternative for HPC
- The software tools ecosystem is **already in good shape**
  - **All encountered issues but one are not specific to Arm platforms**
- The full Isambard XC50 Arm system is coming up now, we're aiming to have early results to share at SC18
- The Arm ecosystem is the **best opportunity for real co-design**

For more information

Comparative Benchmarking of the First Generation of HPC-Optimised Arm Processors on Isambard

S. McIntosh-Smith, J. Price, T. Deakin and A. Poenaru, CUG 2018, Stockholm

http://uob-hpc.github.io/2018/05/23/CUG18.html

Bristol HPC group: https://uob-hpc.github.io/

Isambard: http://gw4.ac.uk/isambard/

Build and run scripts: https://github.com/UoB-HPC/benchmarks

University of BRISTOL

EPSRC

GW4