Pre-exascale Architectures: ARM Performance and Usability Assessment for French Scientific Community

G. Hautreux (GENCI)
Technological watch group,
GENCI-CEA-CNRS-INRIA and French Universities

GENCI

Presentation

- In charge of national HPC strategy for civil research
  - **~14 Pflops available on 3 national centers (CINES, IDRIS and TGCC)**

- Partnerships at the regional level
  - Equip@meso, 15 partners

- Represent France in the **PRACE** research infrastructure

- Promote the use of supercomputing for the benefit of French scientific communities and for industries
  - Specific action to SMEs through the Simseo intiative

TECHNOLOGICAL WATCH GROUP

Led by GENCI and its partners

- **Goals**:
  - anticipate upcoming (pre) exascale architectures
  - deploy prototypes and prepare our users
  - organise code modernization
  - share and mutualise expertise

TECHNOLOGICAL WATCH GROUP

Gather French HPC Expertise

- Early 2017: Deployment of 2 first early technology platforms
  - A Bull Sequana platform at CINES based on Intel® Xeon Phi processor (Knights Landing)
    - 48 nodes => 146 Tflop/s peak performance
    - Infiniband EDR
  - An IBM OpenPOWER platform at IDRIS based on P8+ Nvidia GPU
    - 4*P100 GPU (latest generation) per node
    - Nvlink between P8 and GPU
    - 12 nodes => 254 Tflop/s peak performance (GPU only)

- Third platform currently deployed
  - A Bull Sequana platform at CEA/TGCC based on ARM ThunderX2 processors
    - 30 nodes => 30 Tflop/s peak performance
    - Infiniband EDR

CEA TESTBED: INTI

ARM based prototype

- ARM platform @ CEA/TGCC, Bruyères-le-Châtel (France)
  - 30 thunderX2 nodes (only 6 nodes currently deployed)
    - 2 ThunderX2 CPU (32 cores per CPU @ 2200MHz, 32MB L3 cache)
    - 16 DDR4 DIMM slots (8 channels per CPU)
    - 256GB of memory per node
    - ~ 1 Tflops per node
    - IB Mellanox EDR Interconnect
  - 1920 cores: ~30Tflops
- Software stack deployed by CEA with high level support
  - Bull SCS5 based on RedHat 7.5
  - Compilers
    - ARM compiler v18.4.1
    - ARM compiler v18.4.2
    - GNU compiler v7.3.0
  - OpenMPI v2.0.4 & v3.1.2
- High level support (ARM, ATOS, CEA, GENCI)
  - First workshop organized late September
  - Second workshop schedule for beginning of 2019

MARVELL
CAVIUM NETWORKS

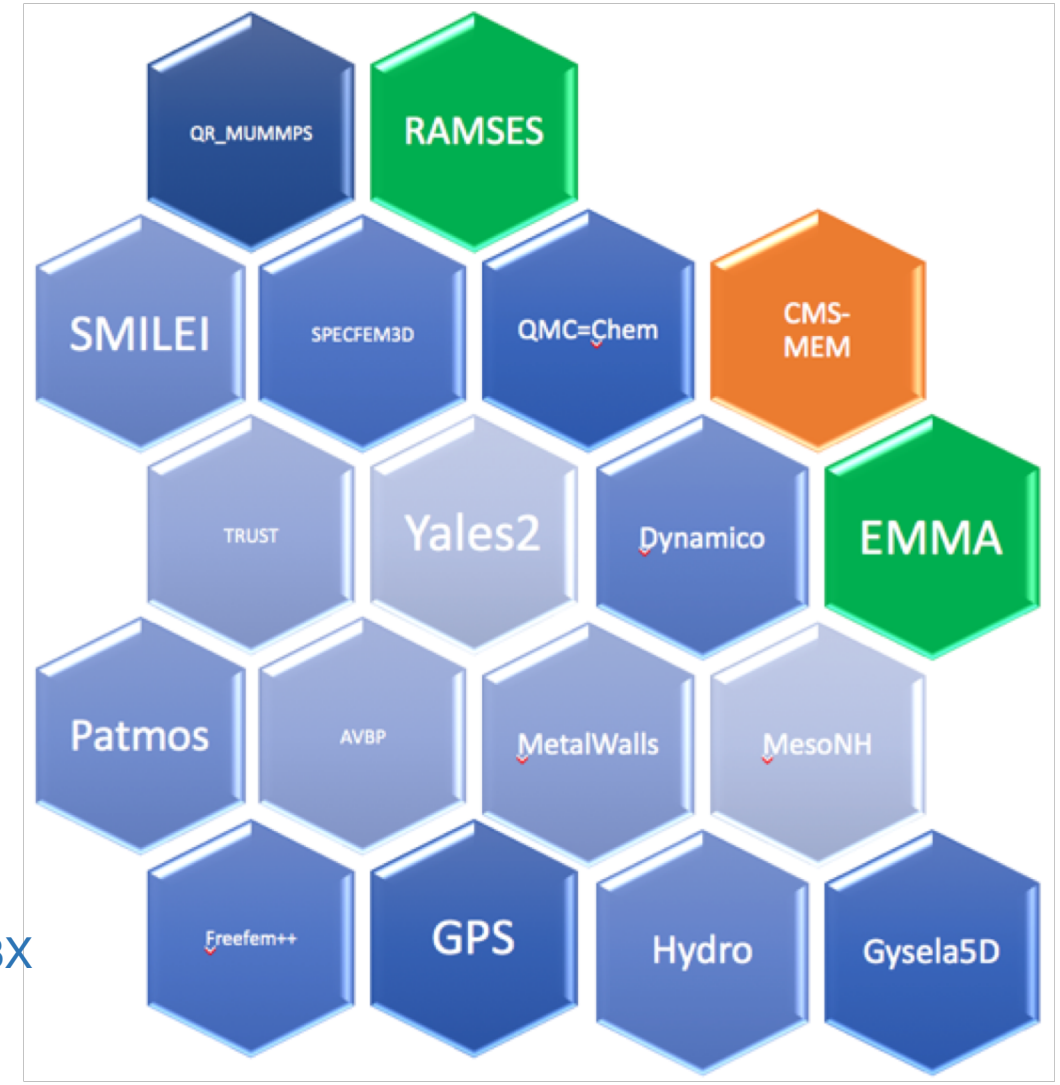
Bull atos technologies arm

RELEVANT SET OF APPLICATIONS

Represent French research community

- 18 « real » applications
  - 2 GPU focused (RAMSES end EMMA)
  - 1 OpenCL (CMS-MEM)
  - 15 « standard applications » coming from various scientific and industrial domains

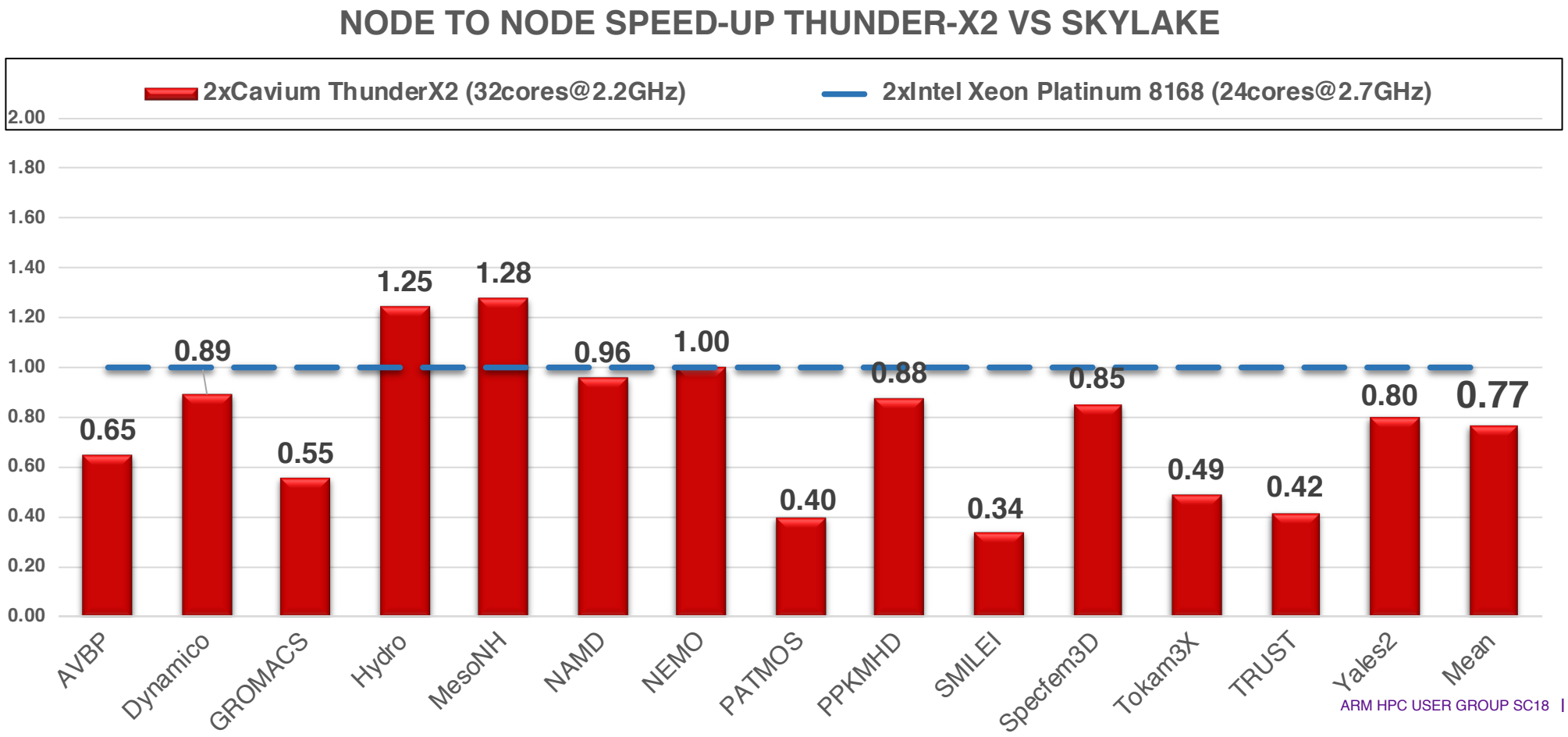
- Inti :
  - Work performed on a subset of 9 apps
    - AVBP, Dynamico, Hydro, MesoNH, PATMOS, SMILEI, Specfem3D, TRUST, Yales2
  - 5 others apps added
    - NAMD, GROMACS, NEMO, PPKMHD, TOKAM3X
  - Results available for 14 applications



PERFORMANCE SUMMARY

Preliminary results

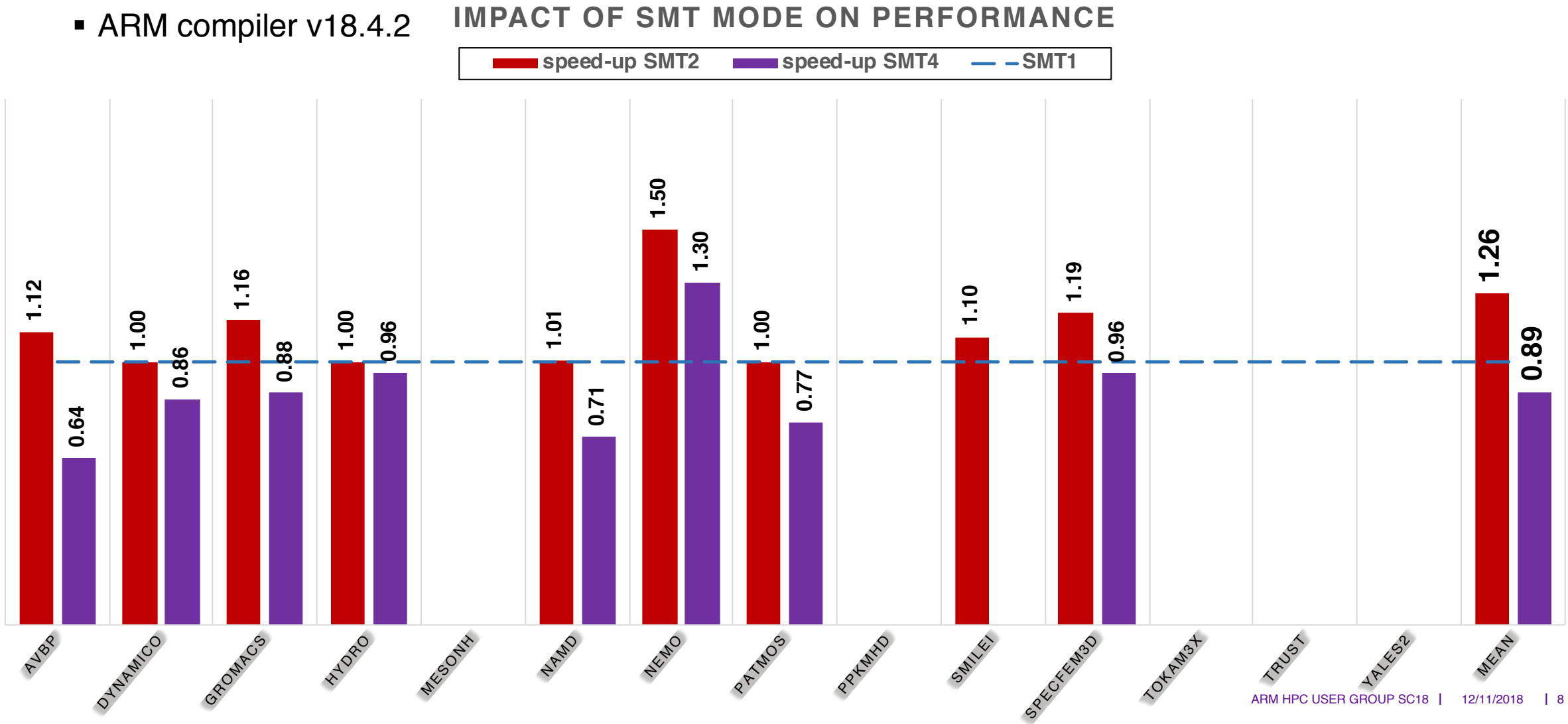
- The overall performance for those applications at the moment:
  - Comparison to Tier-0 machine Irene Joliot-Curie @ CEA/TGCC, Bruyères-le-Châtel (France)
  - ARM compiler v18.4.2 vs Intel compiler 18.x



PERFORMANCE SUMMARY

Preliminary results

- SMT Modes evaluation
  - ARM compiler v18.4.2



FIRST CONCLUSIONS

❑ARM ecosystem

- Ready for HPC ?
  - Portability is not an issue
  - No issue with flang front-end for our applications
  - All 14 applications ported in 0.5 days
- Good performances
  - Compared to top bin Skylake nodes
    - peak performance gap
    - Catalog price gap
    - Reduced TDP for ThunderX2
  - ***Only node to node result at the moment***
- Debugger and profiler available
  - DDT : functional
  - MAP: ease of use, lack of vectorization info
- Libraries available
  - Math libraries up and running

❑ARM platform has to be tuned

- SMT2 seems to be the best
  - Best performances recorded
- SMT4 gives very bad performances
  - Even if we use the exact same number of MPI processes / OpenMP threads as in SMT1 or SMT2
  - Investigate and understand why?
- We are looking forward the ARM compiler v19.x

❑ Still a few improvements to make

- Some applications are better using GNU compiler
  - Two of them have ~10% improvement
- Minor changes in applications to compile
  - E.g. OpenMP reduction perf issues
- Erf() not efficient compared to Intel version
- Please add explicit vectorization information in MAP!

Thank you for your attention!

Questions?

gabriel.hautreux@genci.fr