Software Performance, Portability and Precision

- Sketches of landscape
- IN2P3 project(s)
- Technical highlights
- Advances & concerns

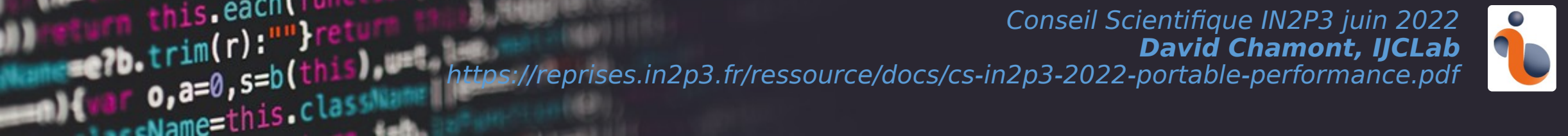
On behalf of the whole Reprises team, with specific contributions from : Pierre Aubert, Arnaud Beck, David Chamont, Hadrien Grasland, Vincent Lafage, Bogdan Vulpescu.

Conseil Scientifique IN2P3 juin 2022
David Chamont, IJCLab
https://reprises.in2p3.fr/ressource/docs/cs-in2p3-2022-portable-performance.pdf

Sketches of landscape

- Challenge
- HEP initiatives
- Performance portability in USA
- Europe computing
- France heading for exascale

Conseil Scientifique IN2P3 juin 2022
David Chamont, IJCLab
https://reprises.in2p3.fr/ressource/docs/cs-in2p3-2022-portable-performance.pdf



Challenge

- We face heterogeneous hardware, especially within our multi-site grids.
- The computing cores within a server are multiplying, specializing, delegating tasks to accelerators.
- The development of code with portable performance becomes a complex challenge.

HEP initiatives

- HEP Software Fundation
  - Working groups : frameworks, software developer tools and packaging, training...
  - HSF WLCG Virtual Workshop on New Architectures, Portability, and Sustainability (2020).
  - Compute Accelerator Forum.
- AidaInnova WP12 : software for future detectors.

Performance portability in USA

- DOE project on performance portability
- P3HPC workshop, run as part of the Supercomputing (SC) program since 2018.

Europe computing

- 2016 : European Cloud Initiative
  - Data infrastructure -> EOSC
  - High-speed connectivity -> GEANT
  - High performance computers -> EuroHPC
- EuroHPC Research & Innovation (~10^9€)
  - HPC Centers of Excellence...
- EuroHPC Infrastructure (~$2x10^9$€)
  - 1st exascale supercomputer @ Jülich

France heading for exascale

- Plan to apply end 2022 for the 2nd european exascale computer. Hardware : CPUs ARM + GPUs.
- Currently : CINES (74.5 Pflops/s), IDRIS (32.3), TGCC (22).
- Necessary to list target domains and applications : aim of workgroup SP3.
- IN2P3 lines up in “sciences of the universe”.
- The workgroup SUN (Sciences et Usage du Numerique) aims to bring out the computational science community
- Achille's heel : GPU porting of applications.

IN2P3 Project(s)

- Decalog, **Reprises**, ComputeOps...
- General goals
- Resources
- Activities

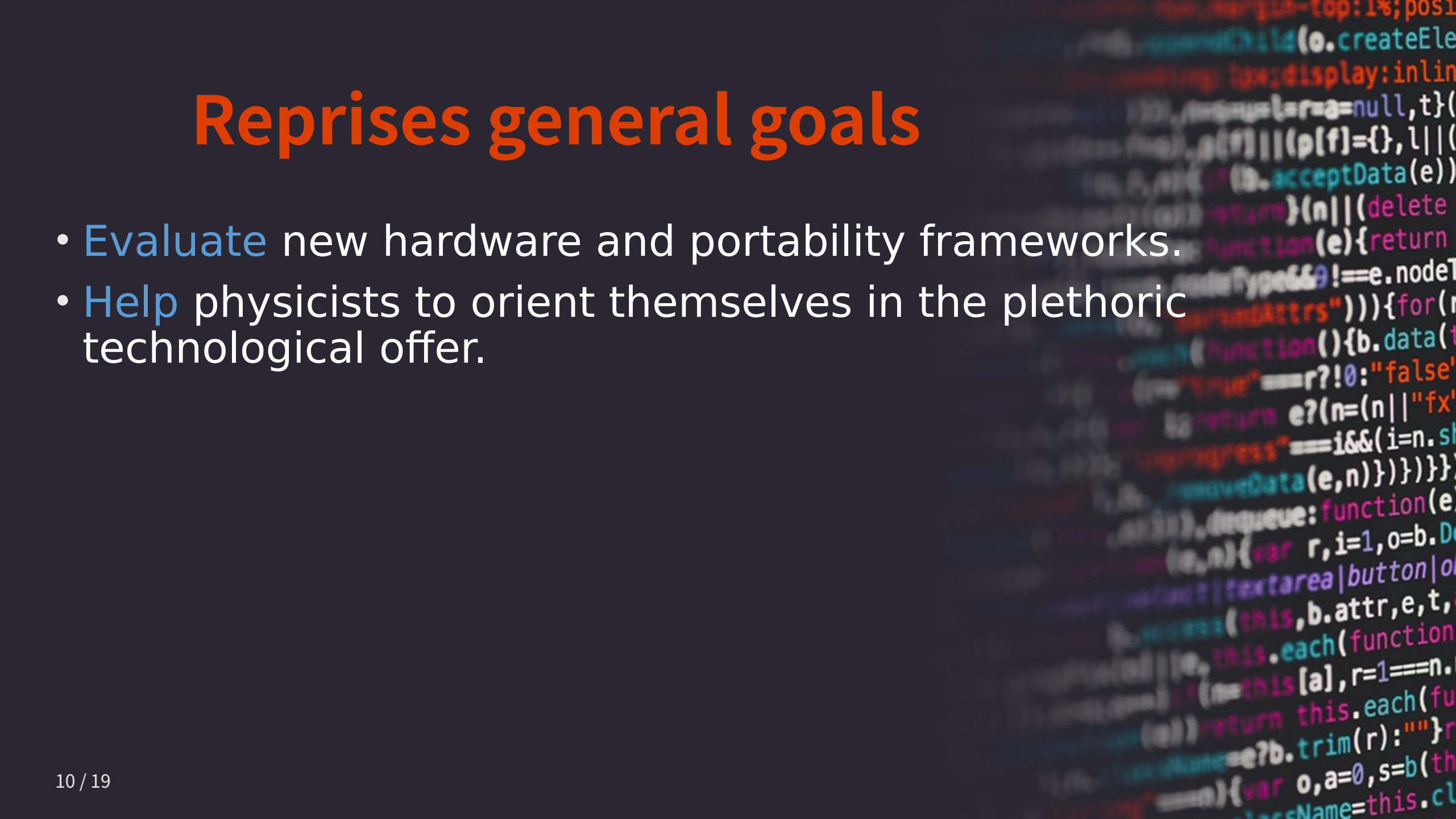
Conseil Scientifique IN2P3 juin 2022
David Chamont, IJCLab
https://reprises.in2p3.fr/ressource/docs/cs-in2p3-2022-portable-performance.pdf

Decalog master-project

- Guideline: software refundation for efficient use of modern hardware
- Two subprojects :
  - Reprises: portabibility, precision, perenity,...
  - ComputeOps: containers for the computing.

Reprises general goals

- Evaluate new hardware and portability frameworks.
- Help physicists to orient themselves in the plethoric technological offer.



Reprises resources

- ~10 engineers, ~1.5 FTE.
- From IJCLAB, IPHC, L2I, LAPP, LLR, LPC, LPNHE, LUPM, SUBATECH.
- ~6 k€ for missions.

Reprises activities

- Internal exchange of feedbacks, through monthly meetings,
- Strong contribution to the institute's prospective, becoming a web guide.
- First demonstration of a server dedicated to scientific computing teaching.
- Collaborations : CEA, EDF, now computer science laboratories...

Co-supervised doctorates

- LUPM 2019-2022: Optimization of the simulation of atmospheric cascades for gamma-ray astronomy experiments. With Philippe Langlois (LIRMM).
- IJCLab 2021-2024: GPU and performance portability - heterogeneous approaches and applications. Funded by UPSaclay. With Joel Falcou (LISN).
- IJCLab 2021-2024: Configuration and control of the accuracy of the calculation, application to low-energy gamma-ray measurements. MITI funding. With Fabienne Jezequelle (LIP6).
-

Technical highlights

- CPU vector registers
- Accelerators (GPU & FPGA)
- Precision & acuracy

Conseil Scientifique IN2P3 juin 2022
David Chamont, IJCLab
https://reprises.in2p3.fr/ressource/docs/cs-in2p3-2022-portable-performance.pdf

CPU vector registers

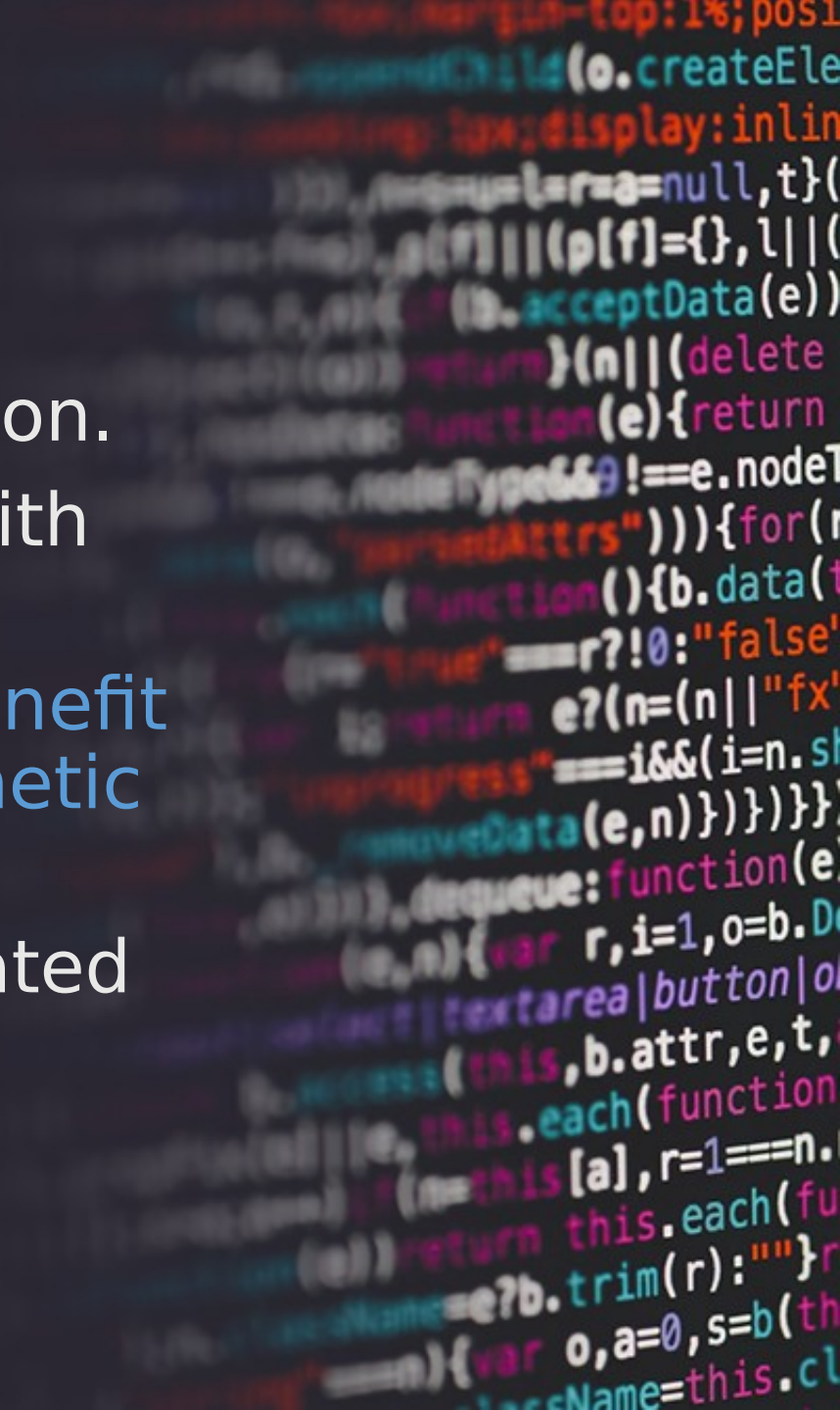
- CPUs can perform up to 16 floating point operation per core at the same time.
- More than 93% of the archivable peak performance.
- Requires appropriate data structure and data parallel computations.
- Many hardware flavors, many software offers, many deployment issues.

Accelerators (GPU (& FPGA))

- NVidia ecosystem stays ahead
- Advocates of less proprietary solution now rather focus on SYCL, very C++ centric.
- HPC community still bets on directives with OpenMP/OpenACC.
- FPGA: harsh competition between Intel and Xilinx, today (too) focused on expensive solutions dedicated to machine learning.

Precision & acuracy

- Using only double is not any more an option.
- Mixing different precisions can be done with generic programming.
- Control of the acuracy is vital, and can benefit from french expertise in stochastic arithmetic tools.
- This problematic is generally underestimated by physicists.



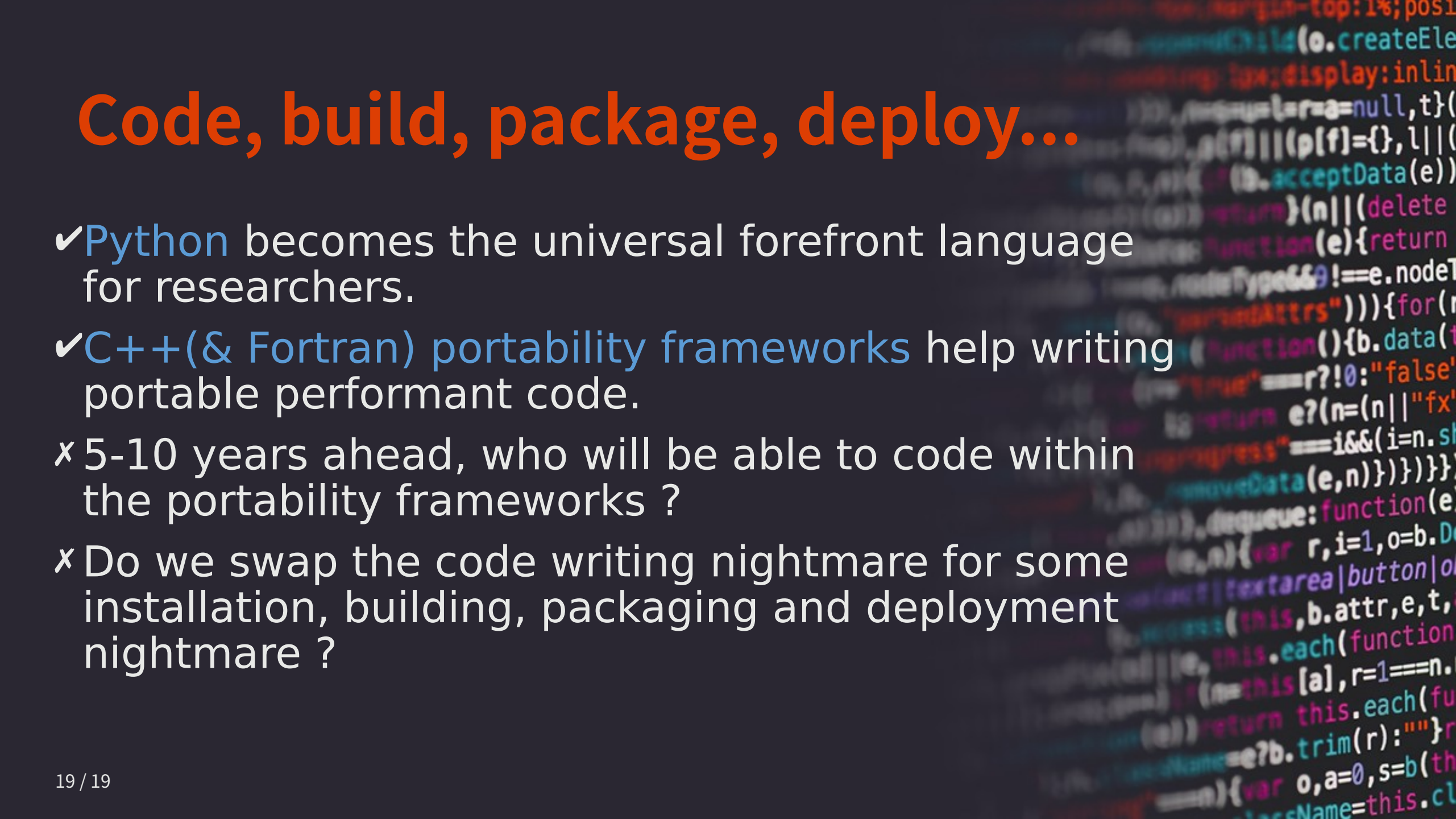
Advances & concerns

- Code, build, package & deploy
- IN2P3 expertise

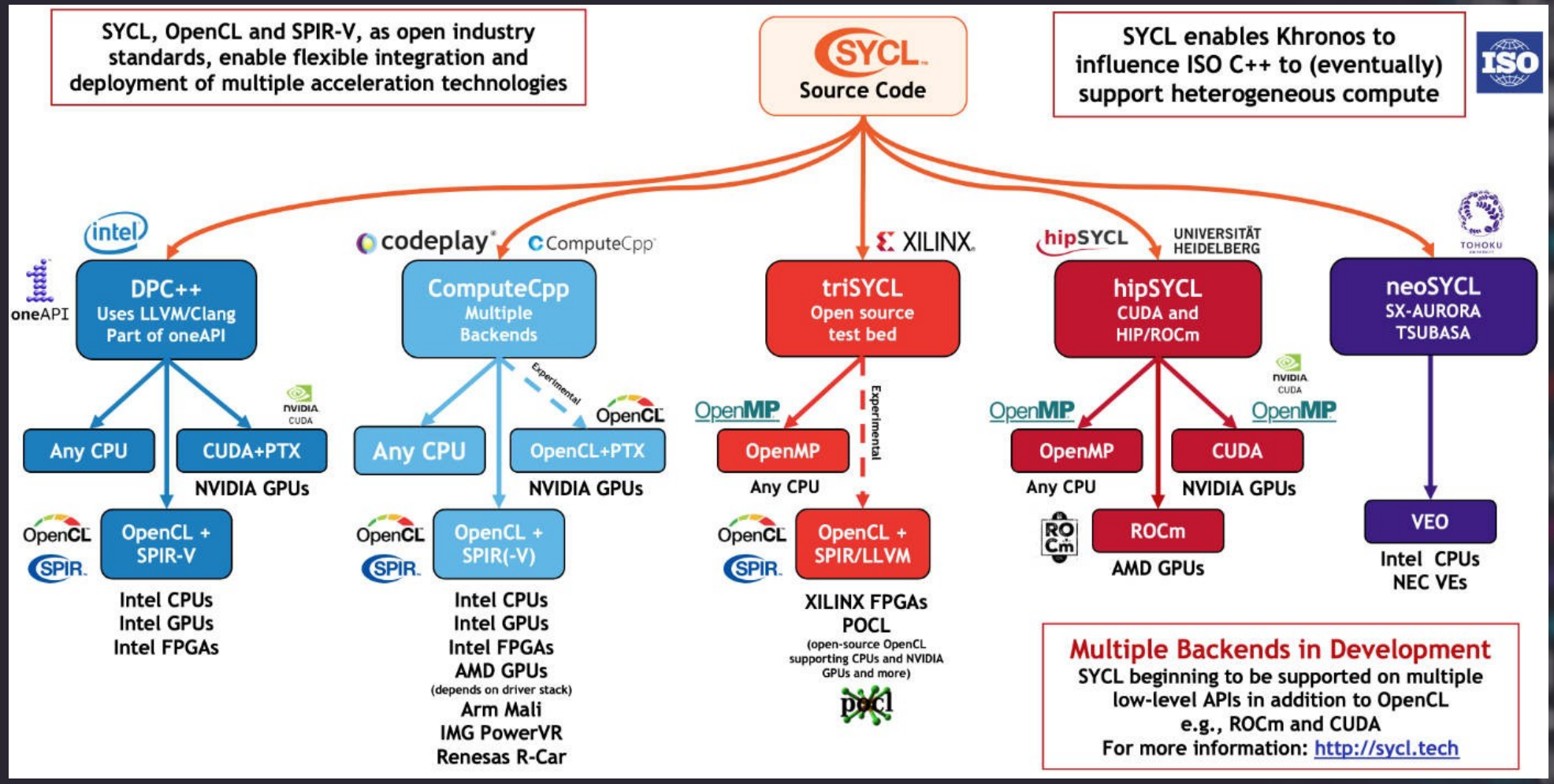
Conseil Scientifique IN2P3 juin 2022
David Chamont, IJCLab
https://reprises.in2p3.fr/ressource/docs/cs-in2p3-2022-portable-performance.pdf

Code, build, package, deploy...

- ✔ Python becomes the universal forefront language for researchers.
- ✔ C++(& Fortran) portability frameworks help writing portable performant code.
- ✗ 5-10 years ahead, who will be able to code within the portability frameworks ?
- ✗ Do we swap the code writing nightmare for some installation, building, packaging and deployment nightmare ?



SYCL


SYCL, OpenCL and SPIR-V, as open industry standards, enable flexible integration and deployment of multiple acceleration technologies
SYCL Source Code
SYCL enables Khronos to influence ISO C++ to (eventually) support heterogeneous compute
ISO
intel
oneAPI
DPC++
Uses LLVM/Clang
Part of oneAPI
Any CPU
CUDA+PTX
NVIDIA GPUs
OpenCL
SPIR
OpenCL + SPIR-V
Intel CPUs
Intel GPUs
Intel FPGAs
codeplay
ComputeCpp
ComputeCpp
Multiple Backends
Experimental
Any CPU
OpenCL+PTX
NVIDIA GPUs
OpenCL
SPIR
OpenCL + SPIR(-V)
Intel CPUs
Intel GPUs
Intel FPGAs
AMD GPUs
(depends on driver stack)
Arm Mali
IMG PowerVR
Renesas R-Car
XILINX
triSYCL
Open source test bed
OpenMP
Experimental
OpenMP
Any CPU
OpenCL
SPIR
OpenCL + SPIR/LLVM
XILINX FPGAs
POCL
(open-source OpenCL supporting CPUs and NVIDIA GPUs and more)
pocl
hipSYCL
UNIVERSITÄT HEIDELBERG
hipSYCL
CUDA and HIP/ROCm
NVIDIA CUDA
OpenMP
OpenMP
Any CPU
CUDA
NVIDIA GPUs
OpenMP
RO Cm
ROCm
AMD GPUs
TOHOKU
neoSYCL
SX-AURORA TSUBASA
VEO
Intel CPUs
NEC VEs
Multiple Backends in Development
SYCL beginning to be supported on multiple low-level APIs in addition to OpenCL
e.g., ROCm and CUDA
For more information: http://sycl.tech

IN2P3 expertise

- ✔New computer science researcher at LAPP
- ✔Improved relationships with computer science
- ✗ Relaxed relationships with computational science
- ? How to consolidate those relationships (*teach -> fund internship -> cosupervise phd)*
- ? How to buy the next Grace Hopper NVidia card (*lack of recurrent budget)*
- ? Enough GPU, packaging, precision specialists...
- ? How to group the experts into an operational tele-team (*how to reward partner labs ?)*

Questions ?

Backup

Conseil Scientifique IN2P3 juin 2022
David Chamont, IJCLab
https://reprises.in2p3.fr/ressource/docs/cs-in2p3-2022-portable-performance.pdf

Serveur ambulant

Matériel, dans une valise “tout-terrain”, 6k€

- CPU Intel Core i9-10980XE (18 coeurs)
- 128 Go de RAM (8x16Go)
- GPU NVidia Quadro RTX 5000
- Borne wifi supportant 30 utilisateurs

Logiciel

- Partitionnement en 2 coeurs interactifs + 16 coeurs benchmark
- Gestion de jobs par slurm
- Outils de monitoring système et benchmarking

HPC Proxy performance+flexibilité

- La compilation de code vectorisé dépend des capacités du matériel ciblé : sse, sse2, sse3, sse4.1, sse4.2, AVX, AVX2, AVX512 ...
- En général, on compile son application une seule fois, pour un niveau de vectorisation donné (le moins-disant des matériels ciblés).
- HPC Proxy permet de générer une bibliotèque intermédiaire, permettant de choisir à l'exécution l'implémentation adaptée au matériel courant.

Plan du guide

1) Si l'application existe, la profiler globalement
2) Choisir une technologie adaptée au problème traité
3) Rendre la précision configurable, et s'efforcer de la réduire
4) Structurer les données en tableaux
5) Privilégier l'algorithme qui passe le mieux à l'échelle
6) Exprimer les noyaux de calcul dans un style fonctionnel