JENA Computing Initiative: Software Challenges, Best Practices and Recommendations

Graeme Stewart, Paul Laycock,
Adrien Matta

ICFA Data Lifecycles Panel, 2024-12-03

JENAA
Joint ECFA-NuPECC-APPEC Activities

JENA Computing Initiative

- What is JENA?
  - Body that identifies synergies and opportunities that would be common between ECFA (European Committee for Future Accelerators - HEP), NuPPEC (Nuclear Physics European Collaboration Committee) and APPEC (Astroparticle Physics European Consortium)
  - Organised by the respective chairs of ECFA, NuPPEC and APPEC
- JENA Computing Initiative
  - 2022 symposium there was a focused session on software and computing
    - It became clear that a more comprehensive identification of computing needs for the next decade, across communities, was needed
    - Particularly the *funding agencies* asked us to prepare some joint view
- There was a follow up seminar in Bologna in 2023
  - This reviewed the situation and formed a series of working groups to formulate a white paper for the next JENA Symposium

Working Groups and Timeline

- WP1 - HPCs
- WP2 - **Software and Heterogeneous Architectures**
- WP3 - Federated Data Management, Virtual Research Environments and FAIR/Open Data
- WP4 - Machine Learning and Artificial Intelligence
- WP5 - Training, Dissemination, Education

The timeline is for all the groups to produce draft reports by the end of November 2024, which will be synthesised into a combined white paper at the start of 2025.

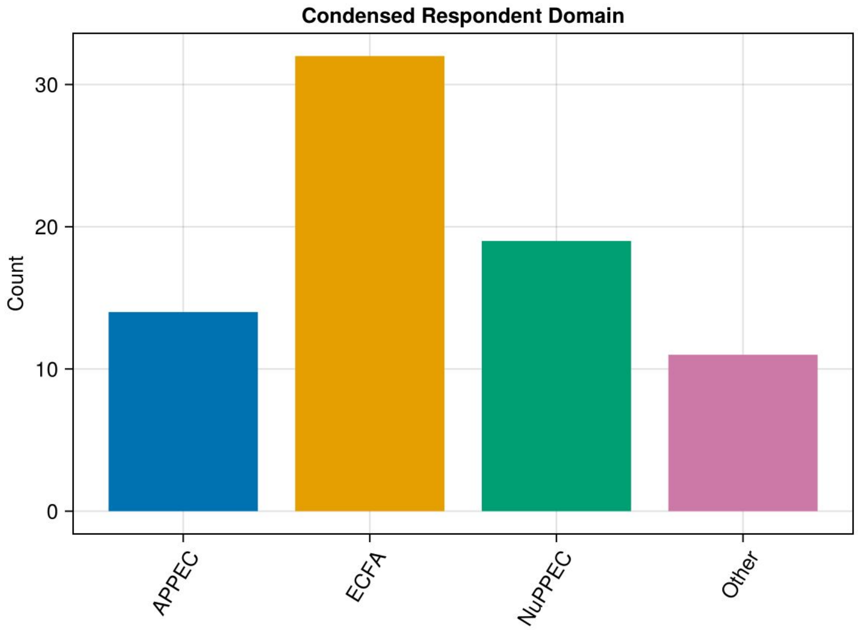
- Mostly these draft reports are close to ready

WP2 - Software and Heterogeneous Computing

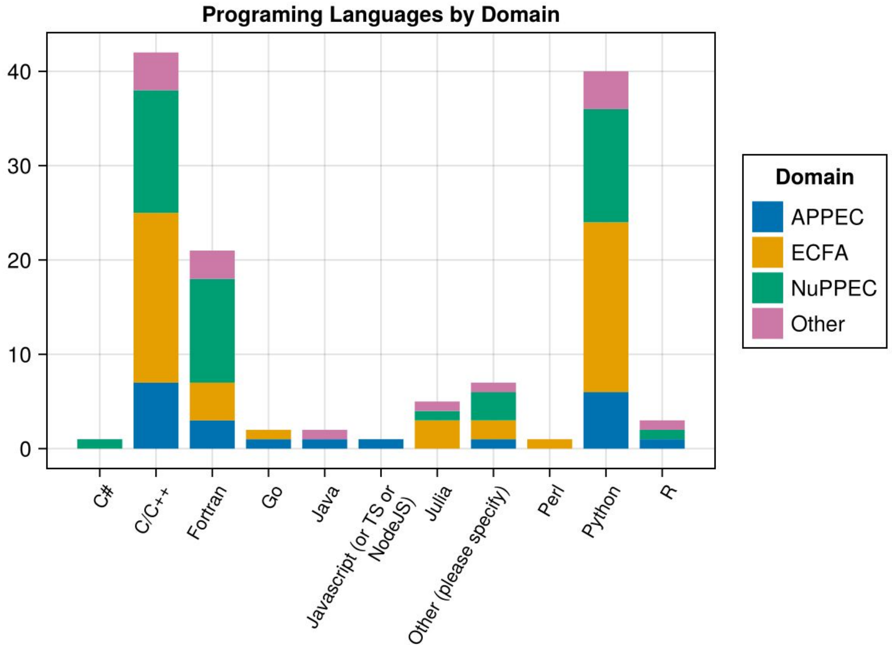
- Formed of ~15 experts from all three communities
- We firstly identified many documents that had already been prepared by the communities and experiments on future challenges
    - E.g., the HL-LHC upgrade documents from ATLAS and CMS
- We then summarised and synthesised these inputs
    - First into super-summaries, per community
    - Then into our final report
- In parallel, we wanted to survey the communities and gather more information
    - We prepared questions about software in common with our colleagues in the Spectrum project (which covers HEP and radio astronomy)
    - This survey ran from July to September

A Few Survey Snippets

- 76 people in total answered
  - More from ECFA, but reasonable number of responses from APPEC and NuPPEC
- 48 people answered the questions on software
- For a more detailed presentation on the results, see this talk

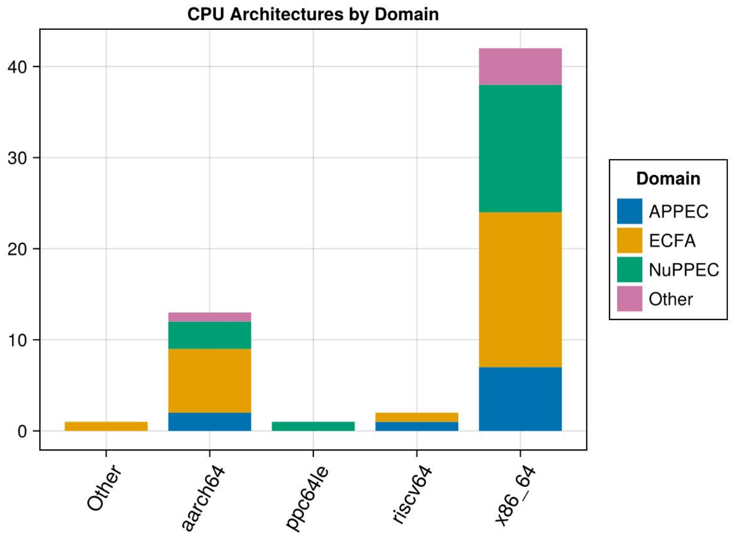


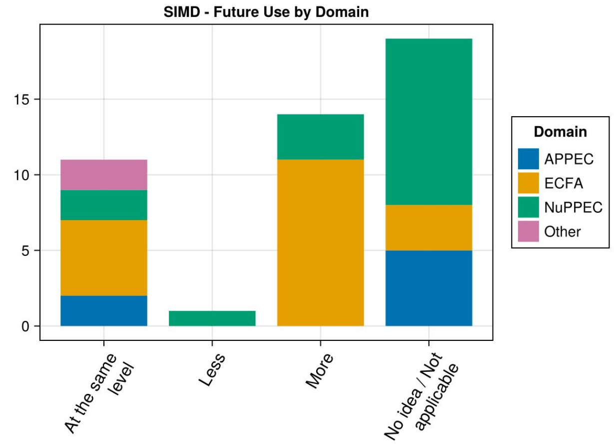
Programming Languages Used

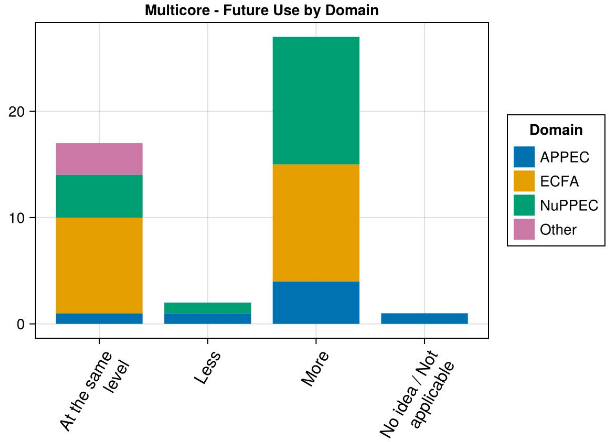


- C++ and Python dominate, as expected
- Fortran had a strong showing
  - Theory codes, particularly in nuclear physics
- Some younger languages are attracting interest
  - Julia, Rust
- Programming accelerators is a hot topic

CPU Architectures, Multicore and SIMD

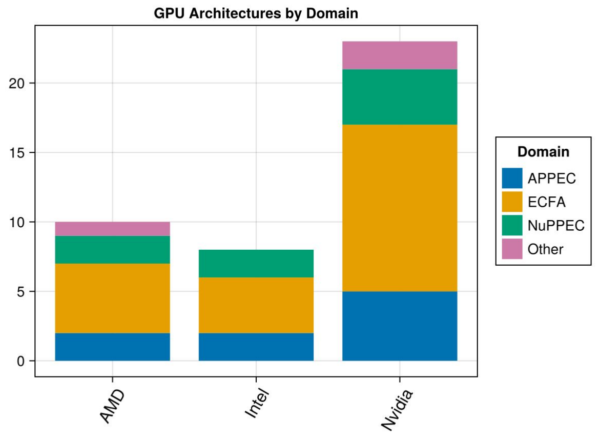


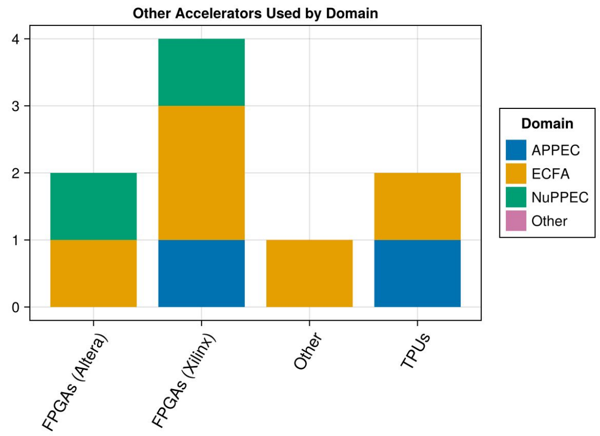




- x86_64 still dominates, strong showing for aarch64 (ARM)
- Expectation is to use more multicore
    - SIMD is much better known in the ECFA community

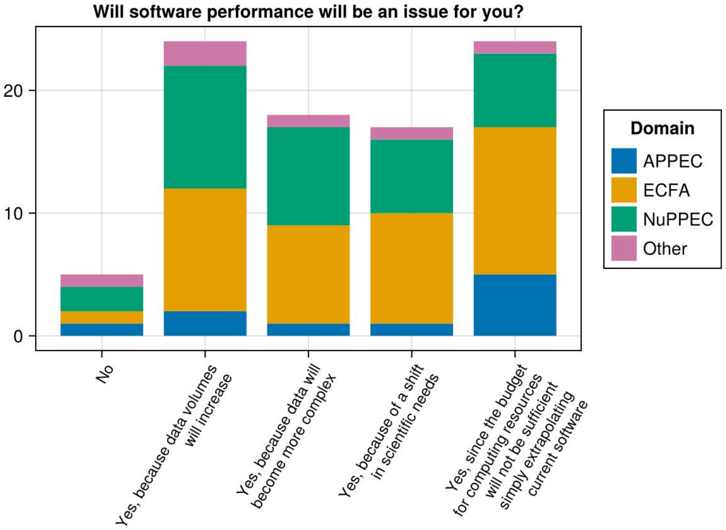
GPUs and other Accelerators

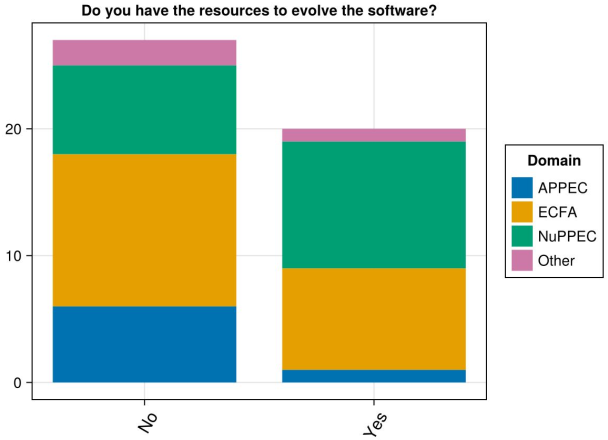


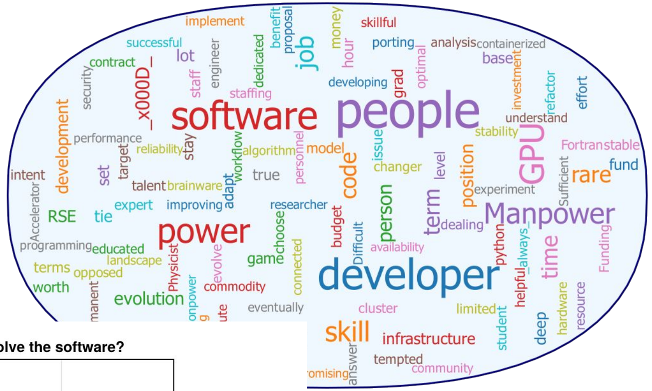


- GPUs are the dominant accelerator, small showing for FPGAs
- There is clear interplay between what centres provide and what processors are targeted by developers
- Direct programming and ad hoc SDKs dominate over heterogeneous toolkits

Software Performance



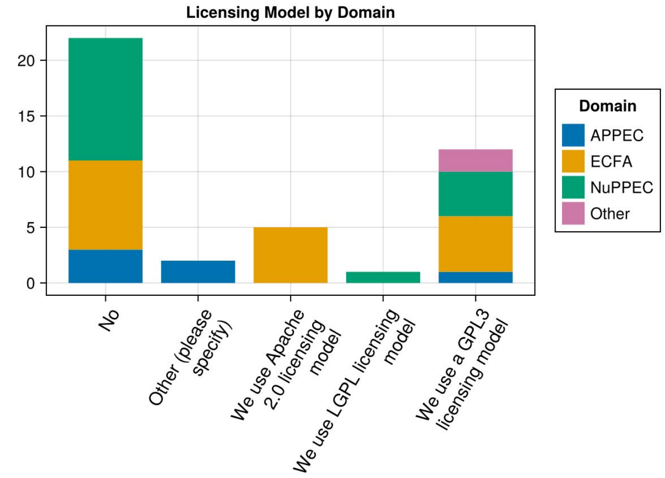


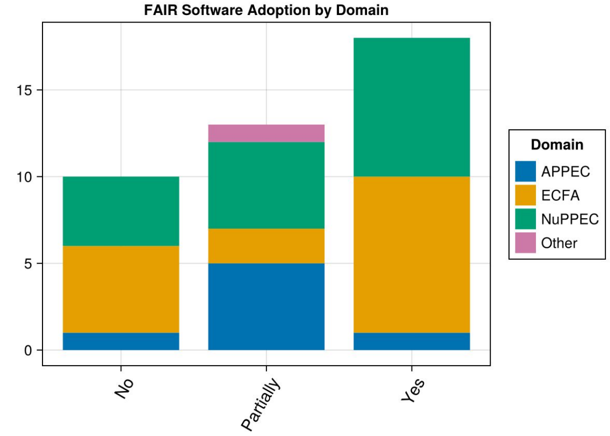


- Software performance is an issue, though resources are partially available
- Missing resources = people!

Licensing and FAIR Software

- A large number of respondents reported having **no** licensing model
  - This is surprisingly high, given this is a basic part of best practice
    - *How can we help?*
- Makes all the more difficult to believe that FAIR4RS is adopted
  - We suspect that people do not actually understand what FAIR4RS means!
  - As “partially” is also a popular response, what causes this? Difficult to understand? Lack of effort? Anachronistic policies?



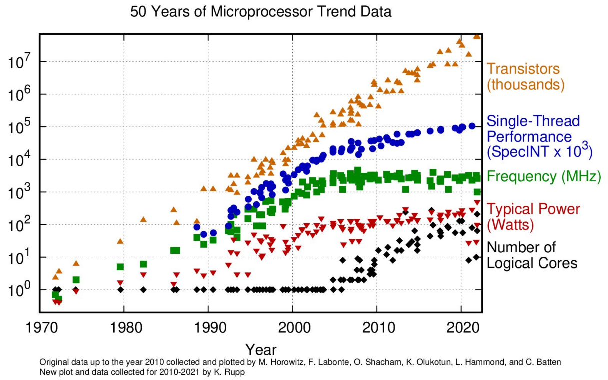


Working Group Report

- This is an advanced draft stage
  - We will release very soon a first public draft

Stefan's talk next

Challenges



- Data volumes grow
  - We are asked to handle our data in a FAIR way
- Computing requirements grow
  - Pushed by both data volumes and physics needs
  - For those communities not yet at the distributed computing scale, it's coming…
- Efficient programming on modern processors is challenging
  - Heterogeneous CPUs with multi-core and SIMD
  - GPUs suit computationally intense uniform problems
  - FPGAs require very specialist skills, but are often needed just after data acquisition
- C++ is fast, but difficult and increasingly rarely known; Python is ubiquitous, accessible, but is very slow
  - Slow computing is bad for the environment!

Best Practices I

- We have identified core best practices, which are well aligned with software development best practices as well as favouring specific needs of our sciences
  - Adopt an open source model, aligned with FAIR4RS, e.g.
    - Ensure software is properly versioned
    - Provide appropriate documentation
    - Deployment and distribution of working software (with dependencies)
  - Automate workflows to improve robustness and minimise resource usage
    - Automation is also key for reproducibility
    - This includes treatment of AI/ML components
      - DevOps + MLOps

Best Practices II

- ...
  - Strive for portable software
    - At least across CPU architectures and OS versions
  - Pay attention to performance for critical code paths
    - This might well mean the use of accelerators
  - Maintain and modernise legacy software
    - Consider rewriting in certain cases (deprecated skills, performance needs)
  - Strengthen software collaborations
    - Reused and widely used software helps minimise efforts
    - This helps extend software lifetimes and improve quality
  - Develop institutional guidelines
    - This is especially valuable for smaller experiments and teams

Recommendations

- We conclude with recommendations to adopt policies, recognising that these also require investment
    - Support Open Science and FAIR principles for software, aligned with robust data preservation and access
    - Invest in software maintenance and in the development of modern software solutions that help reduce environmental impact
    - Strategically invest in software that serves multiple experiments and disciplines and optimises data and workflow management
    - Optimally use and allocate computational resources
    - Recognise new multi-paradigm techniques and invest in training and reward trainers
    - Reward software and computing work and provide suitable career paths

Summary

- JENA Computing Initiative has worked extensively with the HEP, nuclear and astroparticle communities to understand current practice and future needs
  - Touching on key areas such as HPCs, software, distributed computing, AI/ML, and training and careers
- Our reports are in the final drafting stages now
- They will be synthesised into a combined white paper for the community
  - As well as a 10 page executive summary, targeting the funding agencies
- We strongly believe that improved support for software is vital for the scientific outcomes we are striving for
  - That must be reflected in supporting training and careers of scientific software experts