Software production

Second subgroup session

InPEx 2025 workshop – April 15-17, Japan

Our goal today: identify technical challenges & path towards solutions

Some challenges (The What):

- Can we standardize HPC DevOps (HPCOps) methodologies and practices to enhance productivity and enforce inte and portability?
- How to tackle different responsibility models in various computing center?
- How to support CI for the environment of each supercomputer?
- How can we empower users with synergetic co-deployment of > What interfaces can we define to ensure portability and i

A path towards solutions (The How):

- There are already man
  - pack tainer,
  - (ReFrame, Jube, Testpilot,
- W
  -
  -
  -
  -
- The v ooks great, for CI, application packaging
  - identify what prevents every site from providing a virtual-`$site`?
- Containers could be the one interface with the the various sites
  - Sites could provide base images with pre-built applications & libraries if they want
  - There would have to be variants for each HW variation, wow to choose which one to use? => we need to identify the variations required and to normalize system description to support automated choice
  - The question of the interface with system libraries is not easy either (MPI, GPU, etc.), similar questions arise

And now, today: how to build, organize, rationalize a worldwide community to tackle these challenges

Planning of the session – Organizational challenges

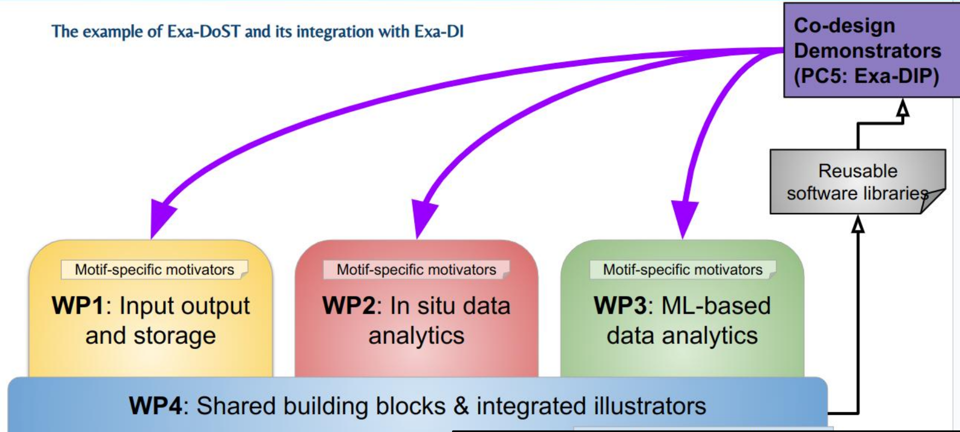
- *“Software production in NumPEx”* (Julien Bigot)
- *“Software production in ECP”* (Todd Gamblin)
- *“Software production for Fugaku”* (Kento Sato)
- *“High Energy Physics, and its* features *(or problems)”* (Tommaso Boccali)
- *“the High Performance Software Foundation”* (Todd Gamblin)
- Open discussion
  - Synergetic user/administrator software build & deployment for HPC
  - Package managers for HPC
  - Containerization & virtualization at exascale
  - Software binary repositories and caches
  - CI/CD/CB at exascale
  - Everything-as-code for HPC
  - Cloudification of the supercomputers
  - AI: new software stacks and practices
  - Shared metadata catalogs of HPC software
  - Methodologies and tools for reproducibility in HPC
  - **HPC software sustainability: funding, communities, foundations, …**

Previous session

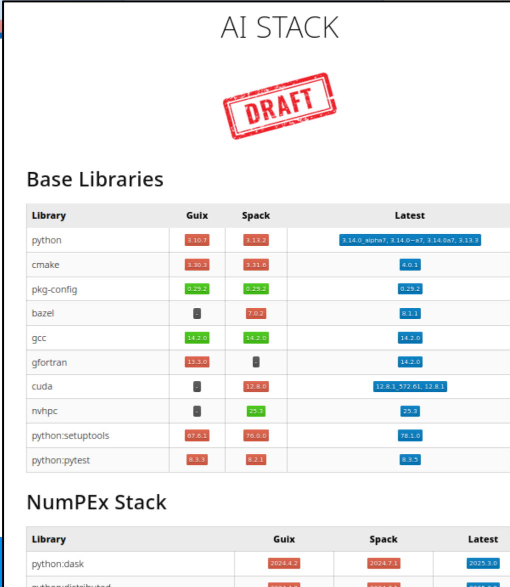
https://tinyurl.com/inpex-swprod

Software production in NumPEx – Julien Bigot

- NumPEx is structured in 5 projects
- Projects 1 to 3 produce reusable technical tools & libraries
- Project 5, Exa-DI is responsible for implementing the interface with applications, by
  - field testing solutions, identifying needs
  - productizing solutions, working on packaging, deployment, CI, etc.
- The strategy for these latest aspect is
  - To empower users beyond modules, with spack & guix
  - To curate a list of software and provide software integration guidelines
    - Identify a list of factual badges self-assessed by developers


The example of Exa-DoST and its integration with Exa-DI

Co-design Demonstrators (PC5: Exa-DIP)
Reusable software libraries
Motif-specific motivators
WP1: Input output and storage
Motif-specific motivators
WP2: In situ data analytics
Motif-specific motivators
WP3: ML-based data analytics
WP4: Shared building blocks & integrated illustrators


AI STACK

DRAFT

Base Libraries

Library	Guix	Spack	Latest
python	3.10.7	3.13.2	3.14.0_alpha7, 3.14.0–a7, 3.14.0a7, 3.13.3
cmake	3.30.3	3.31.6	4.0.1
pkg-config	0.29.2	0.29.2	0.29.2
bazel	-	7.0.2	8.1.1
gcc	14.2.0	14.2.0	14.2.0
gfortran	13.3.0	-	14.2.0
cuda	-	12.8.0	12.8.1_572.61, 12.8.1
nvhpc	-	25.3	25.3
python:setuptools	67.6.1	76.0.0	78.1.0
python:pytest	8.3.3	8.2.1	8.3.5

NumPEx Stack

Library	Guix	Spack	Latest
python:dask	2024.4.2	2024.7.1	2025.3.0
python:distributed	[illegible]	[illegible]	[illegible]

Software production in ECP – Todd Gamblin

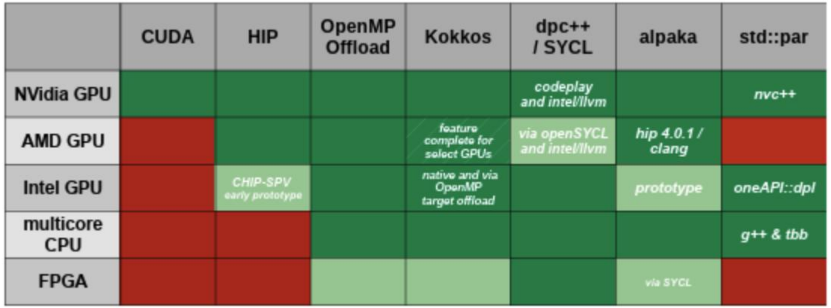
- Big focus: make sure that applications were ready for the HW platforms
- For the 1st time maybe, app developers were convinced they had to use abstraction layers & shared dependencies
- Tracking interest for libraries by applications, milestones in the project related to application integration, made sure that software was actually useful for applications
- Some libs and tools ended up being really used, Kokkos, spack, etc.
- Hierarchy of integration: product, SDKs, E4S
    - SDKs were here to ensure that libraries would actually work together and not assume they were alone
    - Small groups are required to achieve actual work, grouping by theme might not be the only and best solution all the time, but it had the merit to exist
- A database of all products, scraped out of the project webpage
- There was an attempt to establish community policies
- There was a try to do commercial support for E4S, this doesn’t seem really widespread in the end. It was a request by facilities, but they didn’t really use it in the end.

Software production for Fugaku – Kento Sato

- FugakuNEXT will be CPU + GPU => More specific developments required
    - A list of required software is identified
- On Fugaku, a majority of system software is proprietary software that must be continuously maintained by the vendor.
    - Open-source software can be maintained by different vendors or sub-contractors.
- On FugakuNEXT, Riken-RCCS is the main developer, but there is a document specifying some requirements regarding the result.
- OSS is a strong target for the next generation.
- For the programming environment, compiler usage is tracked on Fugaku. C/C++ was raising vs. Fortran, but it’s not a sustained trend. Fortran support will be critical.
    - This is an application co-designed project “application first”
    - So applications have a strong say in the choices made => requesting a programming language change is not realistic
    - There will have to be a porting assistance.

High Energy Physics, and its features (or problems) – Tommaso Boccali

- Software development @ CERN
- 4 collaborations 5-10MLOCs, ~600 dependencies, complex CI/CD, deployment through CVMFS
  - Many ways to populate the CVMFS
- Some Cuda, experiments with ROCm, oneAPI
- Experiments last ~50years, in such a time-span, everything changes
- How to select abstraction layers that one can rely on for this kind of duration? With support the HPC centers… And a critical user mass.
- Is there interest to work together with the HPC community?



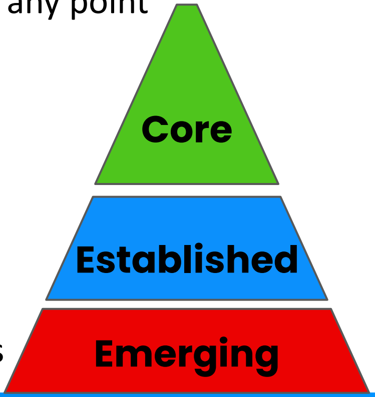
	CUDA	HIP	OpenMP Offload	Kokkos	dpc++ / SYCL	alpaka	std::par
NVidia GPU	🟩	🟩	🟩	🟩	🟩 codeplay and intel/llvm	🟩	🟩 nvc++
AMD GPU	🟥	🟩	🟩	🟩 feature complete for select GPUs	▨ via openSYCL and intel/llvm	🟩 hip 4.0.1 / clang	🟥
Intel GPU	🟥	▨ CHIP-SPV early prototype	🟩	🟩 native and via OpenMP target offload	🟩	▨ prototype	🟩 oneAPI::dpl
multicore CPU	🟥	🟥	🟩	🟩	🟩	🟩	🟩 g++ & tbb
FPGA	🟥	🟥	▨	▨	🟩	▨ via SYCL	🟥

HPSF
HIGH PERFORMANCE SOFTWARE FOUNDATION

hpsf.io

The High Performance Software Foundation – Todd Gamblin

- A neutral hub for open source, HPC software
- Supports projects by
  - Increasing adoption
  - Aiding community growth
  - Enabling development efforts
- Lowering barriers to productive use of today’s & future HPC systems
- Name IP is transferred to the Linux Foundation (the code remains the contributors’ with OSS licence)
  - Ensuring sustainability of software & prevent a specific contributor from closing it at any point
- Accept many projects, doesn’t try to pick “winners”, better to collaborate
- Funded by members
  - This isn’t enough money to pay for developers,
  - but help collaborative efforts, meetings, logos, websites, CI, etc.
- HPSF is organizing the first HPSF conference
- Software stages in HPSF enable to identify the level of reliability, sustainability of projects


Core
Established
Emerging

organizational challenges & path towards solutions

Some challenges (The What):

- How can we improve software sustainability
  - Can we identify critical software pieces in the stack and share the maintenance burden?
  - How to help with software adoption & reuse, vs. redesign from scratch?
- How can we shield these software bits from funding & political instabilities?
- We organize in different way around the world to produce software, how can we combine and collaborate?
- There are many forums that for collaboration around HPC software production & distribution
  - each with their specificities, participants, … => where is the right place to get everyone involved?
  - each with their specific organization => what is their resilience, who do they depend on?

organizational challenges & path towards solutions

Some challenges (The What):

- How can we improve software sustainability
  - Can we identify critical software pieces in the stack and share the maintenance burden?
  - How to help with software adoption & reuse, vs. redesign from scratch?
- How can we shield these software bits from funding & political instabilities?
- We organize in different way around the world to produce software, how can we combine and collaborate?
- **There are many forums that for collaboration around HPC software production & distribution**
  - **each with their specificities, participants, … => where is the right place to get everyone involved?**
  - **each with their specific organization => what is their resilience, who do they depend on?**

A path towards solutions (The How):

- **Cartography the multiple collaboration forums**
  - **their participants**
  - **their organizational dependencies and failure points**
  - **their specificities**

organizational challenges & path towards solutions

Some challenges (The What):

- How can we improve software sustainability
    - **Can we identify critical software pieces in the stack and share the maintenance burden?**
    - **How to help with software adoption & reuse, vs. redesign from scratch?**
- **How can we shield these software bits from funding & political instabilities?**
- We organize in different way around the world to produce software, **how can we combine and collaborate**?
- There are many forums that for collaboration around HPC software production & distribution
    - each with their specificities, participants, … => where is the right place to get everyone involved?
    - each with their specific organization => what is their resilience, who do they depend on?

A path towards solutions (The How):

- **HPSF has interesting features**
    - **A neutral hub place for HPC software development with an independent Legal existence**
    - **A lifecycle with well identified & peer reviewed stages to help identify software maturity**
    - **We will work on evaluation and drafting recommendations for HPSF**

organizational challenges & path towards solutions

Some challenges (The What):

- How can we improve software sustainability
  - Can we identify critical software pieces in the stack and share the maintenance burden?
  - How to help with software adoption & reuse, vs. redesign from scratch?
- How can we shield these software bits from funding & political instabilities?
- We organize in different way around the world to produce software, how can we combine and collaborate?
- There are many forums that for collaboration around HPC software production & distribution
  - each with their specificities, participants, … => where is the right place to get everyone involved?
  - each with their specific organization => what is their resilience, who do they depend on?

A path towards solutions (The How):

- Cartography the multiple collaboration forums
  - their participants
  - their organizational dependencies and failure points
  - their specificities
- HPSF has interesting features
  - A neutral hub place for HPC software development with an independent Legal existence
  - A lifecycle with well identified & peer reviewed stages to help identify software maturity
  - We will work on evaluation and drafting recommendations for HPSF

Concrete outputs & actions planned

- Produce a white paper to formalize the various responsibility models over the world
    - Identify the roles involved, clarify how each model map these roles to actual people, design benchmark use-cases for the various software deployment solutions on each model
- Continue the work initiated to develop an interoperable installed software database format
- Work to standardize containers as the one interface with computing centers
    - Work with computing sites could to provide base images with pre-built applications & libraries
    - Identify a set of required variants and push for a standard site description format
- Generalize the virtual Fugaku approach all over the world
    - Work with Riken to extract a virtual-`$site` framework
    - Work with computing centers over the world to identify requirements for implementation
    - Push for a worldwide deployment
- Cartography the multiple collaboration forums
    - their participants, their organizational dependencies and failure points, their specificities
    - => Maybe organize a BoF @ SC or ISC to gather this community of communities?
- Evaluate and draft recommendations for HPSF from the point of view of all InPex participating institutions

Work will happen on InPEx slack #wg-software-production