International PostExascale

Workshop Series

InPEx 2025 workshop – April 15-17, Japan

InPEx Software production & integration working group

Session 1: technical challenges & solutions

Our goal today: identify technical challenges & path towards solutions

Identify challenges (The What):

- Can we standardize HPC DevOps (HPCOps) methodologies and practices to enhance productivity and enforce interoperability and portability?
- How can we empower users with synergetic co-deployment of the software stack with the administrators of supercomputers?
- What interfaces can we define to ensure portability and interoperability between tools?
- What else?

Identify a path towards solutions (The How):

- Things we can build on
  - package managers (spack, easybuild, guix, …)
  - curated HPC software repositories (E4S, EESSI, guix-HPC, …)
  - containerization (apptainer, singularity, PCOCC, docker, …)
  - CI/CD executors (github actions, gitlab CI, Jenkins, …)
  - Test & benchmark orchestrators (ReFrame, Jube, Testpilot, Pavilion2, …)
- Existing standardized interfaces
  - OCI images & registry
- Interfaces we should standardize
  - installed software database, system firmware, kernel & libraries configuration
- What else?

https://tinyurl.com/inpex-swprod

Reminder of the outputs and milestones from Sitges:

- Can we make the package managers (Spack, Easybuild, Guix) work together?
    - Ongoing spack/easybuild discussions
    - **shared award**: HPCwire Choice Award for Best HPC Programming Tool or Technology
- Submit join paper(s)
    - Testing frameworks, CI/CD (better consensus on the tools used), interactions between different pack-mans ?
    - TODO
- Propose a SC'24 BOF: Empowering users with package managers
    - Not accepted this time :/
- CI/CD
    - Collaboration in anticipation of next EU CI/CD project ?
    - CI/CD at HPSF? Could be a good base for collaborations
- Organize and rationalize a worldwide community
    - Adac/Inpex/JLESC: how to leverage these different discussion groups (does not always include the same people/institutions)?
    - More discussion needed: invite everyone to Inpex slack channel for follow up discussions

Planning of the session

- “*Some context on the challenges*” (Julien Bigot)
- “*Software deployment from a user perspective*” (Lucas Gasparino)
- “*Virtual Fugaku & software deployment status on Fugaku*” (Kento Sato)
- “*Latest news from Spack*” (Todd Gamblin)
- you?
- 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, …

Dedicated session tomorrow

https://tinyurl.com/inpex-swprod

Some context on the challenges

Julien Bigot

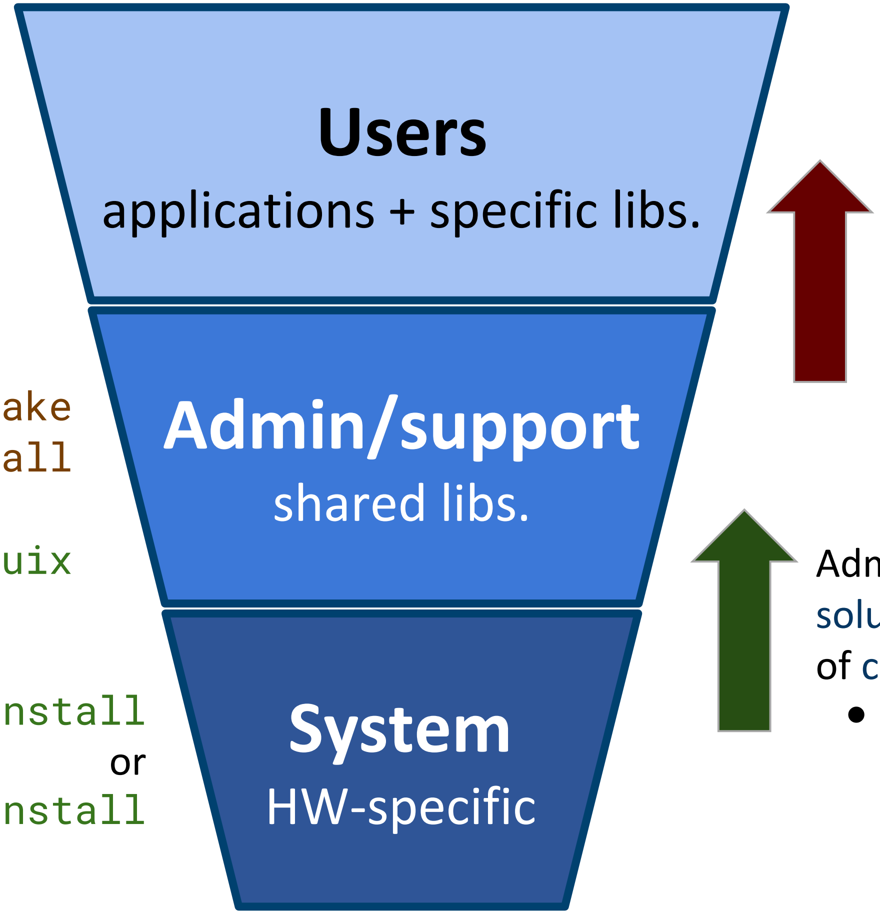
Software deployment on supercomputers: the interface situation

`module load`
+ `./configure` or `cmake`
+ `make` or `ninja`

Historically:
`./configure` or `cmake`
+ `make` or `ninja install`

Now:
`spack` or `easybuild` or `guix`

`sudo dnf install`
or
`sudo apt install`



Module is the historical standard interface

- mostly supports cmake-style on top

Solutions exist when admin & user tools match

- *e.g.* spack `upstream` chaining

No standardized interoperable solution yet

- database format / naming scheme

Admin/support deployment tools have built-in solutions to autodetect a well identified limited set of critical system libraries:

- MPI, GPU, etc.

Software deployment on supercomputers: the challenges from a user perspective

When deploying application software on a supercomputer

- How do you identify the subset of dependencies that are already available and fit your needs?
- How do you know which ones are compatible with each other?
- How do you identify those dependencies you **must** reuse and should really not rebuild by yourself, for example to leverage hardware (interconnect, accelerators, …) to its best.
- Amongst those not available, how do you know which ones to build yourself, which ones to ask from support?
- How do you communicate with support what you actually need and don’t end up with something close en
- How do you automate the build of the remaining pieces of software, in the right order, pointing to the right dependencies, without being root, with a reasonable usage of space & inodes

How to do all of that in a “portable” way, so as not to restart from scratch on each new supercomputer?

- That might use distinct hardware, system distribution, support/admin deployment tool, …

Software deployment on supercomputers: the challenges from a library developer perspective

When producing a library targeted at HPC applications

- In what format should I distribute my library?
  - As binary packages (deb, rpm, pip, binary tarball)?
    - Is there a way to install those on supercomputers without being root?
    - Will the compilation match the environment of the supercomputer?
  - As a container image?
    - How can container images from multiple libraries be combined?
  - As simple source tarballs?
    - What dependencies will already be available on the supercomputer? What amount of transitive dependencies will my user be willing to build? What of dependencies should I vendor?
    - How to build dependencies in cmake, should I really re-implement a package manager in cmake?
  - As source packages/recipes (spack, easybuild, guix, ports, …)?
    - Will my user be able to use it, on a supercomputer, if this is not the native deployment tool?
- How to validate that the library will build and work in the software environment of a given supercomputer?
  - What if the security policy prevents automated connection, limits usage of compute resources?
  - How to test in a software environment that matches the production environment?

Planning of the session

- “*Some context on the challenges*” (Julien Bigot)
- “*Software deployment from a user perspective*” (Lucas Gasparino)
- “*Virtual Fugaku & software deployment status on Fugaku*” (Kento Sato)
- “*Latest news from Spack*” (Todd Gamblin)
- you?
- 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, …

Dedicated session tomorrow

https://tinyurl.com/inpex-swprod

Our goal today: identify technical challenges & path towards solutions

Identify challenges (The What):

- Can we standardize HPC DevOps (HPCOps) methodologies and practices to enhance productivity and enforce interoperability and portability?
- How can we empower users with synergetic co-deployment of the software stack with the administrators of supercomputers?
- What interfaces can we define to ensure portability and interoperability between tools?
- What else?

Identify a path towards solutions (The How):

- Things we can build on
  - package managers (spack, easybuild, guix, …)
  - curated HPC software repositories (E4S, EESSI, guix-HPC, …)
  - containerization (apptainer, singularity, PCOCC, docker, …)
  - CI/CD executors (github actions, gitlab CI, Jenkins, …)
  - Test & benchmark orchestrators (ReFrame, Jube, Testpilot, Pavilion2, …)
- Existing standardized interfaces
  - OCI images & registry
- Interfaces we should standardize
  - installed software database, system firmware, kernel & libraries configuration
- What else?

https://tinyurl.com/inpex-swprod

International PostExascale

Workshop Series

InPEx 2025 workshop – April 15-17, Japan

InPEx Software production & integration working group

Session 2: organizational challenges & solutions

Our goal today: identify organizational challenges & path towards solutions

Identify challenges (The What):

- How can we improve software sustainability
  - How can we identify critical software pieces in the stack and share the maintenance burden at a worldwide level?
  - How can we shield these software bits from funding & political instabilities?
- 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?
- We organize in different way around the world to produce software, how can we combine and collaborate?
- What else?

Identify a path towards solutions (The How):

- Can we identify the multiple collaboration forums
  - their participants
  - their organizational dependencies and failure points
  - their specificities
- Does HPSF answer the needs, does its organization take into account the specificities of each of our organizations?
- Do we need to instantiate new communication channels, mailing lists, slack, … ?
- What else?

https://tinyurl.com/inpex-swprod

Reminder of the outputs and milestones from Sitges:

- Can we make the package managers (Spack, Easybuild, Guix) work together?
    - Ongoing spack/easybuild discussions
    - shared award: HPCwire Choice Award for Best HPC Programming Tool or Technology
- Submit join paper(s)
    - Testing frameworks, CI/CD (better consensus on the tools used), interactions between different pack-mans ?
    - TODO
- Propose a SC'24 BOF: Empowering users with package managers
    - Not accepted this time :/
- CI/CD
    - Collaboration in anticipation of next EU CI/CD project ?
    - CI/CD at HPSF? Could be a good base for collaborations
- Organize and rationalize a worldwide community
    - Adac/Inpex/JLESC: how to leverage these different discussion groups (does not always include the same people/institutions)?
    - More discussion needed: invite everyone to Inpex slack channel for follow up discussions

Planning of the session

- “*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)
- you?
- 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, …**

Yesterday’s session

https://tinyurl.com/inpex-swprod

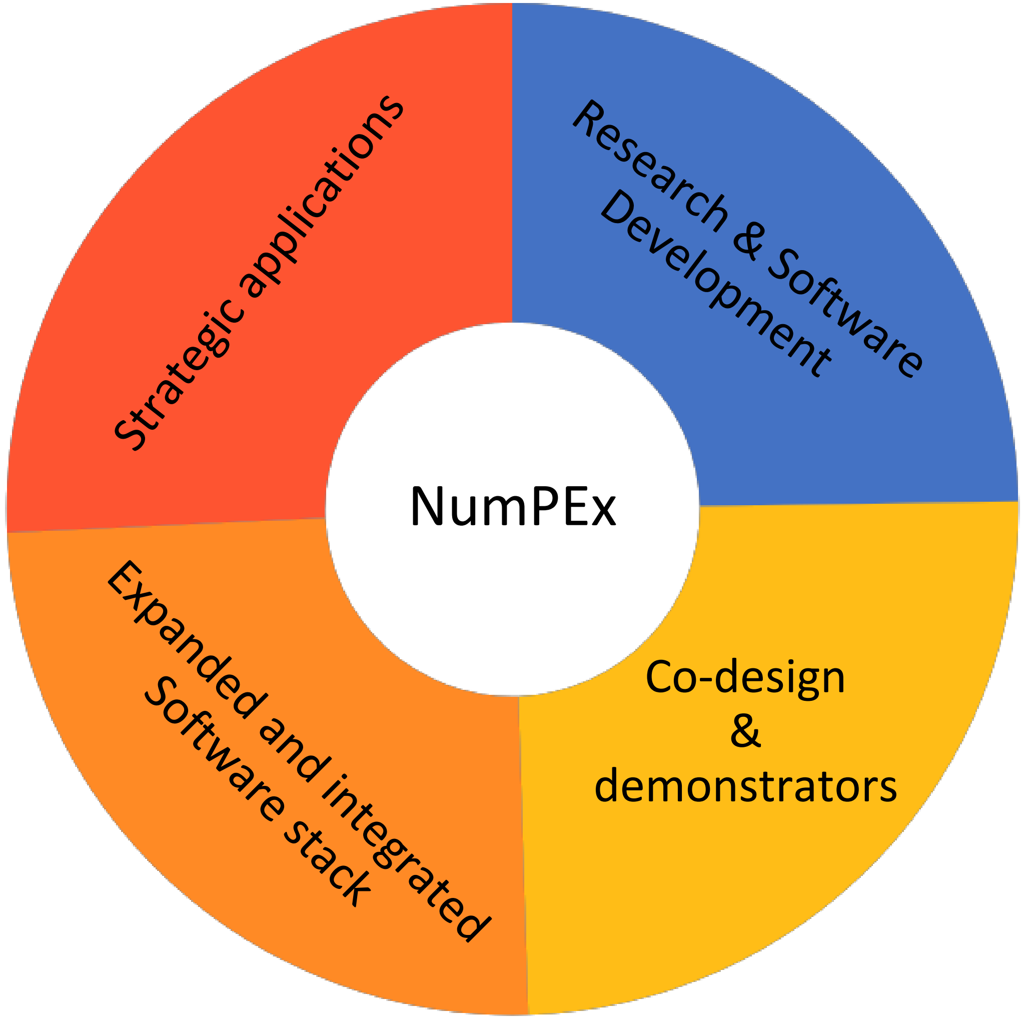
Software production in NumPEx

Julien Bigot

The French NumPEx Program: Approach:

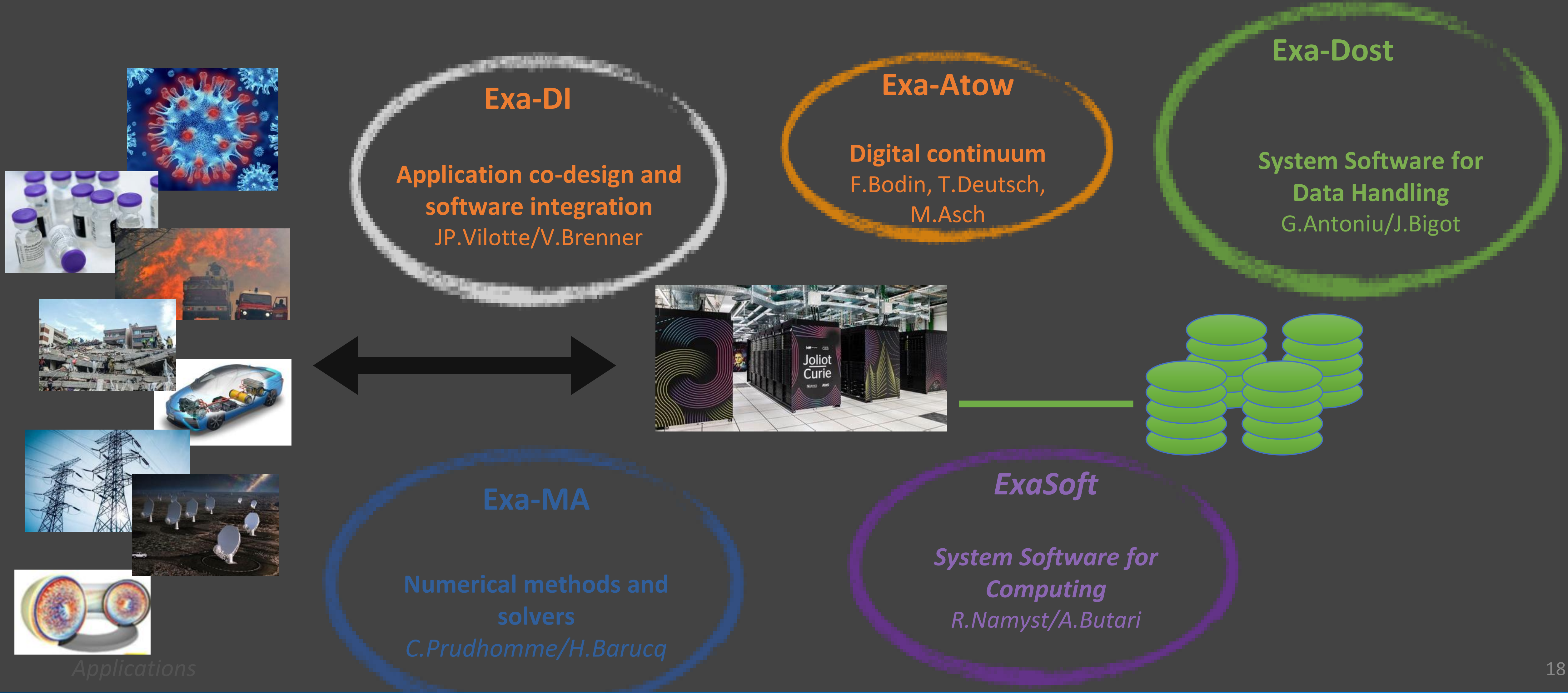
- Prepare for the upcoming European Exascale supercomputers in JSC, Germany & CEA, France
- Consolidating and accelerating the construction of a European exascale software stack and strategic applications exascale capability in a coherent and multi-year framework
- Integrate and validate co-designed innovative methods, libraries and software stack with demonstrators of strategic applications.
- Accelerate science-driven and engineering-driven developers training and software productivity
- Foster national and international collaborations to prepare for the Exascale and post-Exascale era

https://numpex.org


Strategic applications
Research & Software Development
NumPEx
Expanded and integrated Software stack
Co-design & demonstrators

Help aggregate the French Edge/Cloud/HPC/HPDA/IA community

The French NumPEx Program: Workplan



Exa-DI

Application co-design and software integration
JP.Vilotte/V.Brenner

Exa-Atow

Digital continuum
F.Bodin, T.Deutsch, M.Asch

Exa-Dost

System Software for Data Handling
G.Antoniu/J.Bigot

Exa-MA

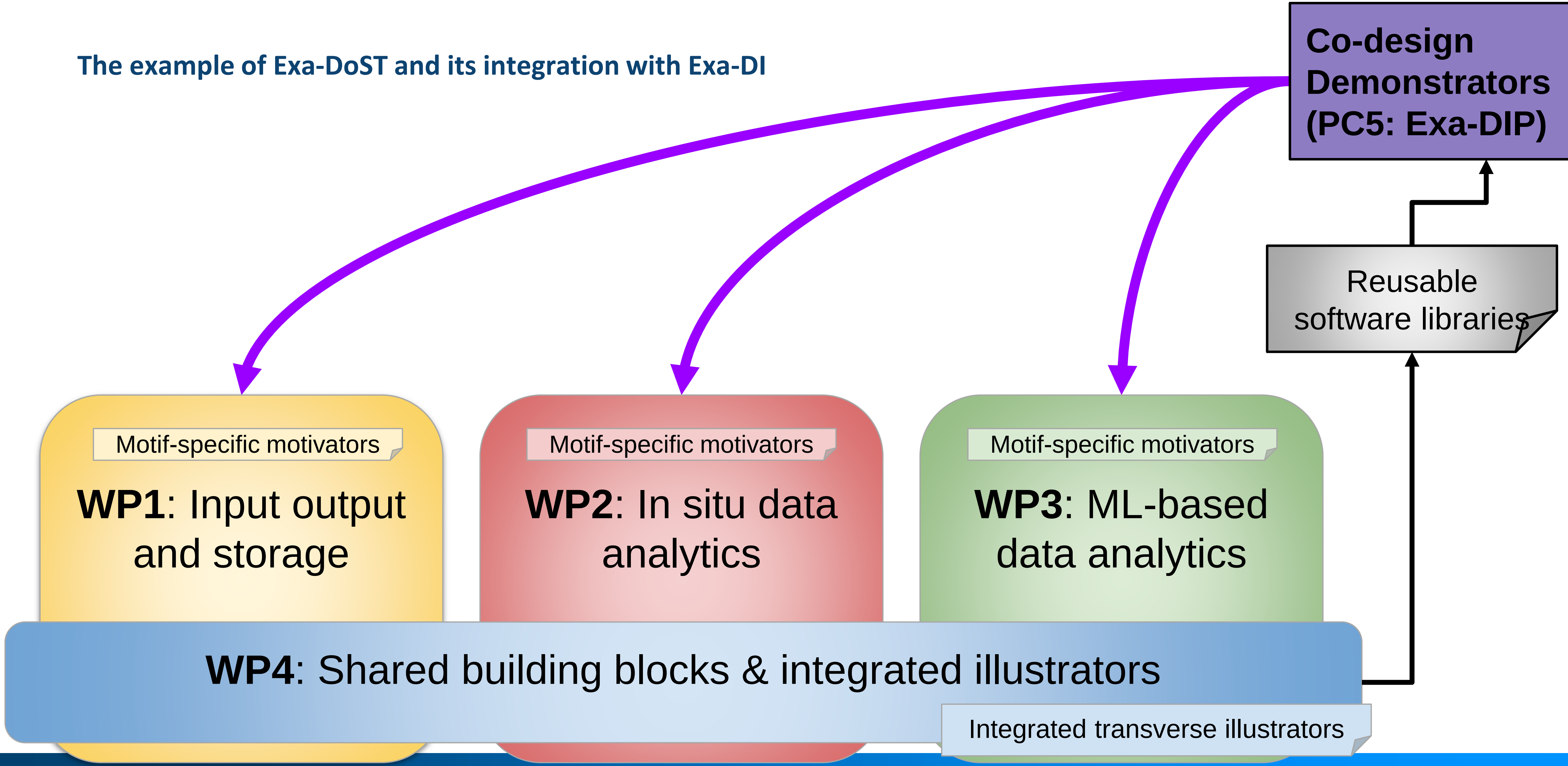
Numerical methods and solvers
C.Prudhomme/H.Barucq

ExaSoft

System Software for Computing
R.Namyst/A.Butari

Applications

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

Integrated transverse illustrators

In Exa-DI: co-Design and software Integration

- Organized in 2 teams
- Compute & Data team
    - ~10 FTEs
    - agile organization
    - works on application motifs illustrated by mini- and proxy- apps
    - integrate, evaluate software produced in NumPEx
    - gives feedback on application needs
- Enabling team
    - ~5 FTEs
    - works on packaging, CI, deployment
    - offers training to the whole of NumPEx
    - improves tooling in synergy with computing center

In Exa-DI : the enabling team

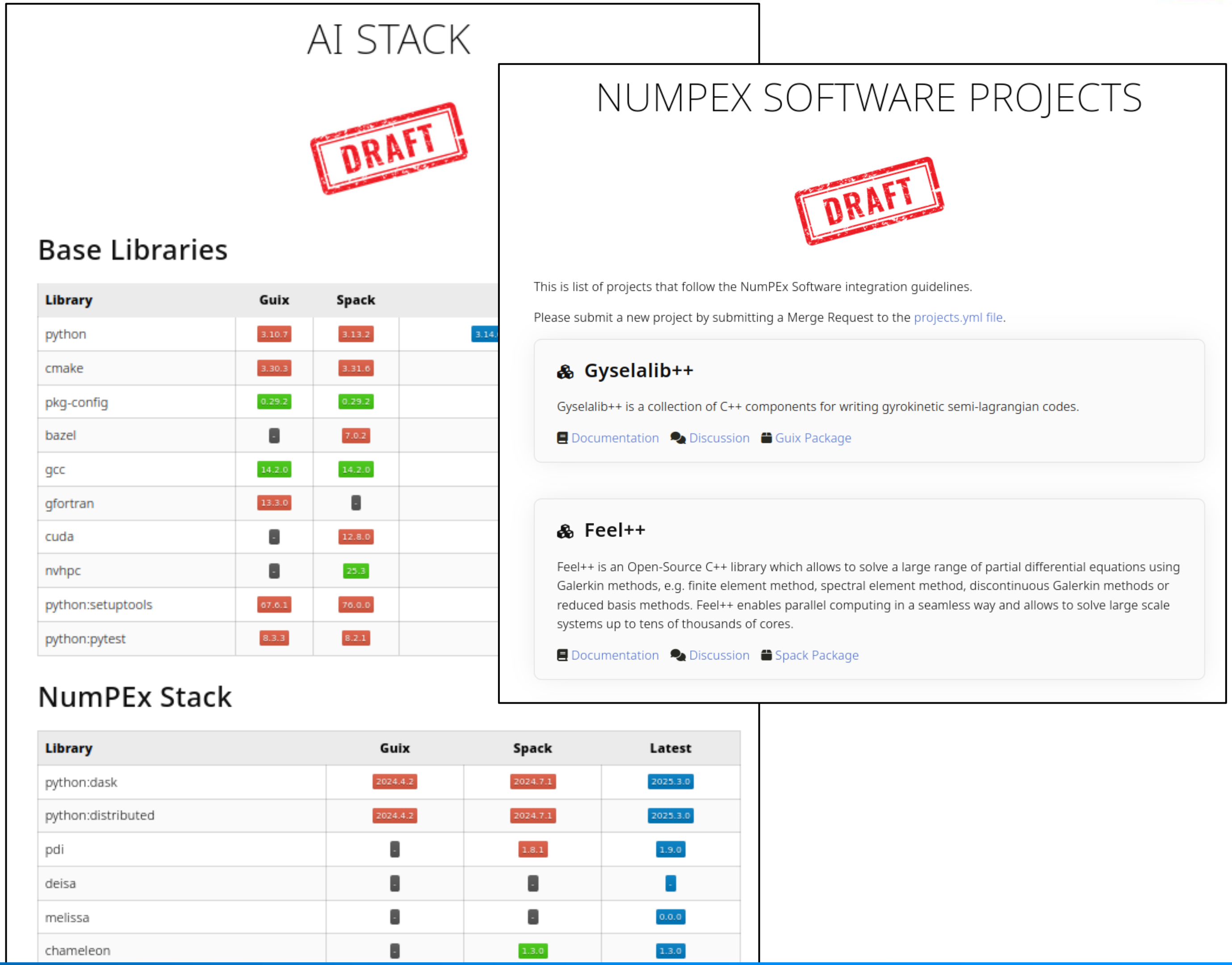
- **Empowering users beyond modules** with modern Package Managers (Guix, Spack)
    - A vision shared within NumPEx!
    - Need more tutorials: spack, guix, cmake
    - Portability: need a description of the low level software system (cuda driver version)
- **Move forward with Software Integration Guidelines and NUMPEX software list**
    - Badges: factual (self-evaluation)
    - versus quality/outreach (peer-evaluation ?)
- Reproducibility in HPC:
    - Target ?
    - Methodology (see reproducibility MOOCs) ?
- Continuous performance benchmarking (CB): still not common practices, but becoming easier:
    - Tools: Reframe, Jube, Ramble
    - EuroHPC initiative to enable runners on some supercomputers (Lummi, Vega)
    - Tradeoff between frequency and energy cost ?
    - Information about "sustained" software scalability ?

NumPEx software production: definition, evaluation and Implementation

NUMPEX Software Integration Guidelines:

1. Packaging (Spack, Guix)
2. Minimal Validation Tests
3. Public Repository
4. Clearly-identified license
5. Minimal Documentation
6. Open Public Discussion Channel
7. Metadata
8. API compatibility information
9. Minimal Performance Tests

Inspired by ECP, to be published online



AI STACK

DRAFT

Base Libraries

Library	Guix	Spack	
python	3.10.7	3.13.2	3.14.
cmake	3.30.3	3.31.6	
pkg-config	0.29.2	0.29.2	
bazel	-	7.0.2	
gcc	14.2.0	14.2.0	
gfortran	13.3.0	-	
cuda	-	12.8.0	
nvhpc	-	25.3	
python:setuptools	67.6.1	76.0.0	
python:pytest	8.3.3	8.2.1	

NumPEx Stack

Library	Guix	Spack	Latest
python:dask	2024.4.2	2024.7.1	2025.3.0
python:distributed	2024.4.2	2024.7.1	2025.3.0
pdi	-	1.8.1	1.9.0
deisa	-	-	-
melissa	-	-	0.0.0
chameleon	-	1.3.0	1.3.0

NUMPEX SOFTWARE PROJECTS

DRAFT

This is list of projects that follow the NumPEx Software integration guidelines.

Please submit a new project by submitting a Merge Request to the projects.yml file.

Gyselalib++

Gyselalib++ is a collection of C++ components for writing gyrokinetic semi-lagrangian codes.

Documentation Discussion Guix Package

Feel++

Feel++ is an Open-Source C++ library which allows to solve a large range of partial differential equations using Galerkin methods, e.g. finite element method, spectral element method, discontinuous Galerkin methods or reduced basis methods. Feel++ enables parallel computing in a seamless way and allows to solve large scale systems up to tens of thousands of cores.

Documentation Discussion Spack Package

Planning of the session

- *"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)
- you?
- 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, …**

Yesterday's session

https://tinyurl.com/inpex-swprod

Our goal today: identify organizational challenges & path towards solutions

Identify challenges (The What):

- How can we improve software sustainability
  - How can we identify critical software pieces in the stack and share the maintenance burden at a worldwide level?
  - How can we shield these software bits from funding & political instabilities?
- 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?
- We organize in different way around the world to produce software, how can we combine and collaborate?
- What else?

Identify a path towards solutions (The How):

- Can we identify the multiple collaboration forums
  - their participants
  - their organizational dependencies and failure points
  - their specificities
- Does HPSF answer the needs, does its organization take into account the specificities of each of our organizations?
- Do we need to instantiate new communication channels, mailing lists, slack, … ?
- What else?

https://tinyurl.com/inpex-swprod