Forum TERATEC 23

Container technologies for HPC

Thierry.Porcher@doit-now.tech

www.doit-now.tech

doitnow

Container Technologies for HPC
Forum TERATEC 23

Thierry PORCHER
Do IT Now – Technical Director
thierry.porcher@doit-now.tech

Alberto GARCIA
Do IT Now – HPC Architect
alberto.garcia@doit-now.tech

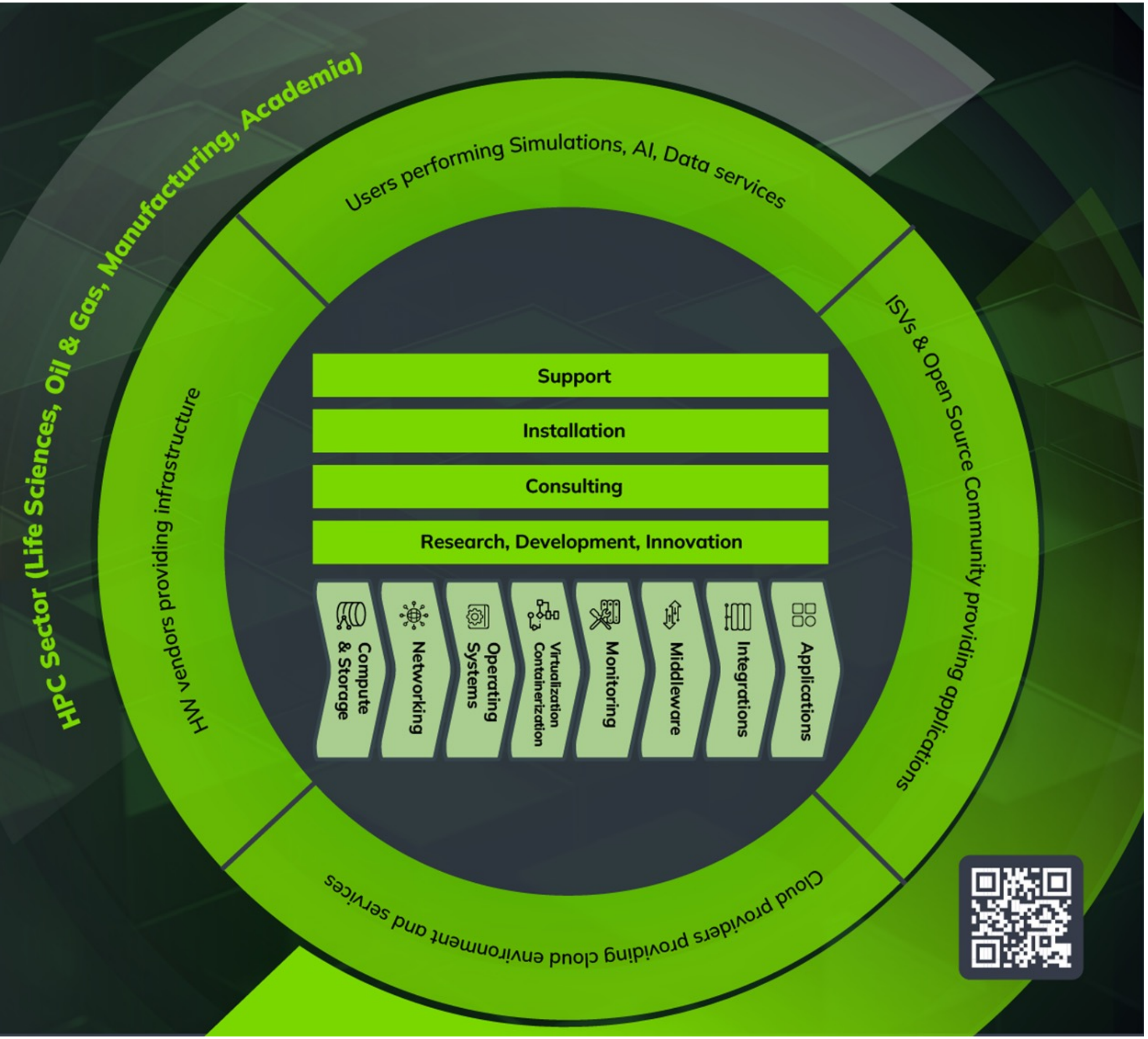
doitnow
HPC Services

Numbers at a Glance

- Active in all of Europe covering WW needs
- 200+ European customers in industry and academy
- 30+ Years of expertise
- 100+ HPC & AI services and solutions professionals
- Managed Services capabilities
  - 12k users/year
  - 150 clusters/year
  - 200+ training sessions/year
- Installations capabilities
  - 5 top500 clusters

Headquarters

- Montpellier (France)
- Barcelona (Spain)
- Turin (Italy)
- Munich (Germany)
- Auckland (New Zealand)
- Switzerland Opening Soon!



What are containers?

- For the traditional HPC user:
    - A way to submit batch jobs using libraries and applications packed in my own image
    - Managing dependencies (applications, version of libraries, OS, …)
- For the AI/new gen user:
    - A way to deploy my own Jupyter/VSCode interactive environment and access GPU resources
    - Up to large scale computations/trainings

Use Case Analysis

- Image administration.
  - How are images stored?
  - What is the format of the images?
  - Are images shared?
  - Is additional infrastructure required?

- Image creation (*user* and *admin*).
  - From live filesystem.
  - From a public HUB.
  - From recipe files.

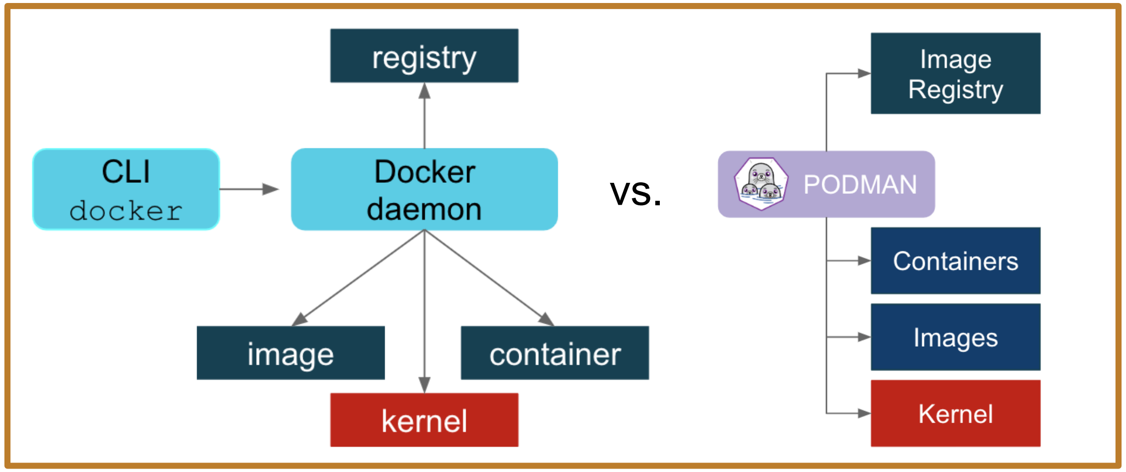
- Image execution on the HPC cluster.
  - Workload manager integration.
  - Parallel file system integration.
  - Accelerator integration.
  - High-performance network integration.

- Extra use cases:
  - [Compatibility with K8s?]
  - Integration with CI/CD tools?
  - Integration with third-party tools (e.g., Artifactory, XRAY, …)?

doitnow

Security Analysis

- Security architectures in container technologies.
  - Root-owned daemon.
  - *setuid* binaries.
  - Full-unprivileged mode



- Security analysis.
  - CVE overview during last years.
  - Releases and updates periodicity during last years.

Other aspects to consider when choosing a technology

Interoperability analysis

- Compliance with OCI image specification.
  - I.e., can the technology use other image formats without conversion?
- Compliance with OCI runtime specification.
  - I.e., can the technology be used by other technologies on top (e.g. K8s).
- Compliance with OCI registry specification.
  - I.e., can the technology download/use containers from public hubs?

License and support

- Type of license.
  - Free or commercial.
- Support models.
  - Type, SLA, and availability.
- Community activity.
  - Non-official support.

Container Tools

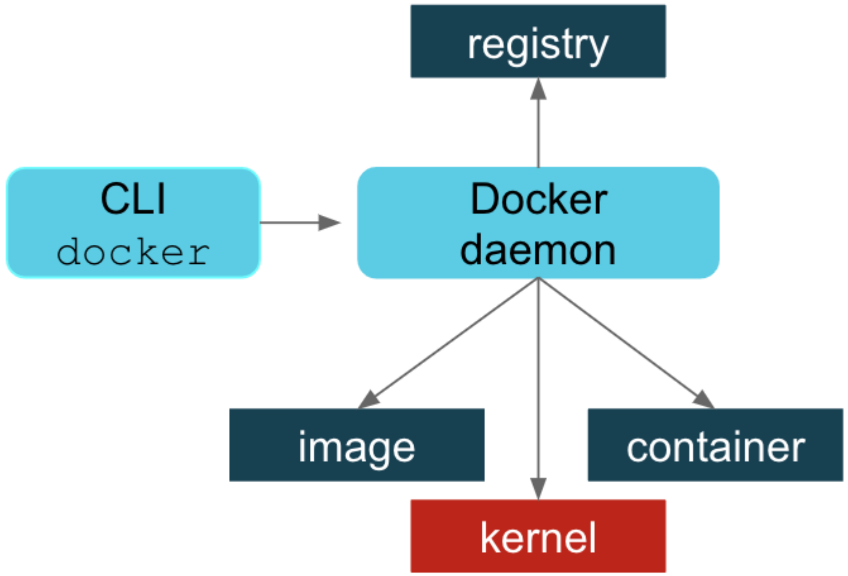
apptainer.org

docker

SARUS

Docker Model

- Architecture
  - Root-owned daemon.
    - Spawns containers.
    - Build images.
    - Pulls/pushes imgs from/to registry.
  - Only root or the docker group can interact with the daemon.
    - Anyone inside docker group can easily escalate to root.
  - User owning the container must be specified.
    - Otherwise, it will be root.



- Security model
  - As root, Docker can leverage the 6 privileged namespaces.
    - Create virtual networks.
    - PID abstraction.
    - Resource limitation (cgroups).
    - Create and mount images.

doitnow

Singularity Model

Architecture

- No daemon.
- Single process execution.
  1. Singularity process launched.
  2. Namespaces setup and checks.
  3. Replace code by the user application (execvp).
- Same users inside and outside the container. No user mapping.

Security model

- Some **setuid** binaries in order to.
  - mount the Singularity container image.
  - create the necessary namespaces in the kernel.
  - bind host paths into the container.
- Full rootless possible with user namespaces.
  - no-setuid version of the binaries.
  - Reduced features.
    - No images.
    - No bind mounts.

Singularity vs. Apptainer

- Singularity code forked in 2020 and now two products coexist.
  - Singularity from Syslabs.
    - https://sylabs.io/docs/
    - SingularityCE but also SingularityPRO and Singularity Enterprise if advanced support is needed.
  - Apptainer which joined the Linux Foundation
    - https://apptainer.org/
    - Official support by CIQ

- Singularity and Apptainer differ fundamentally on the security architecture.
  - Singularity opts for a setuid model and avoids the use of user namespaces.
  - Apptainer opts for full unprivileged model, relying heavily on user namespaces and the newest kernel features.

Sarus Model (CSCS)

SARUS

Idea

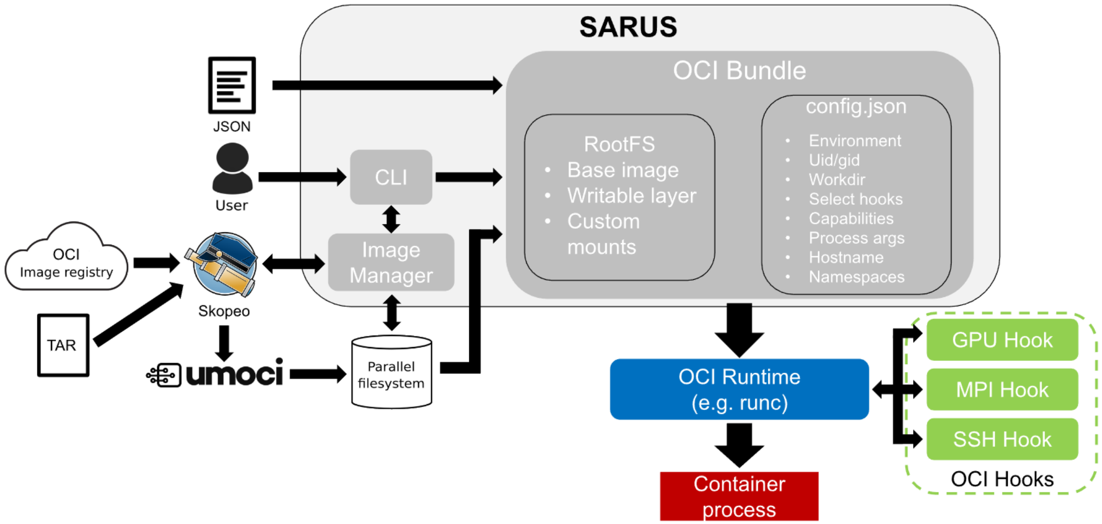
- Container runtime and tools heavily focused on HPC.
  - Batch scheduler, MPI, and GPGPUs integration.
  - Minimal overhead on container spawn.
- Key element: OCI hooks.
  - Custom plugins that allow to interface subsystems natively within the container.
    - MPI
    - Parallel file systems.
    - GPGPUS

Security model

- Setuid binaries.
- No root daemon on target platform.
- No user namespaces.

Architecture

- Extend image with config.json file in order to add metadata.
- Delegate image life-cycle (creation, storage) to docker.
  - Docker registry deployed in an auxiliary platform near to the supercomputer.



Podman Model

Idea

- Provide a replacement of Docker avoiding the root-owned daemon.

Architecture

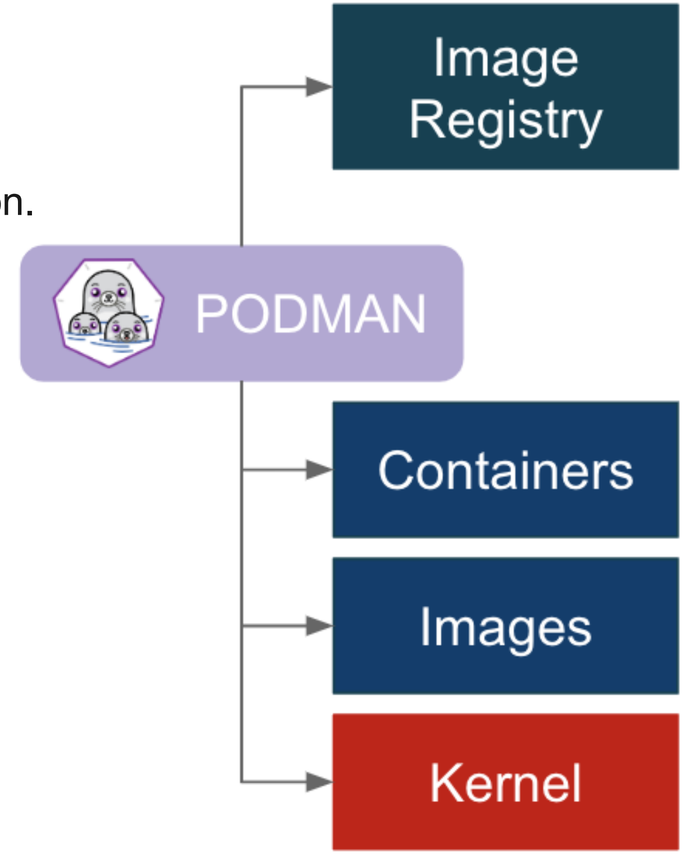
- No daemon, no setuid executables.
- fork/exec model: podman spawns the container.
- Images stored in $HOME

Security model

- Total rootless for basic operations.
  - Spawn a container.
- Execute with sudo when it’s necessary.
  - Build an image.
- Tricks for avoiding privileged operations.
  - FUSE for certain types of mounts (OverlayFS).
  - slirp4netns program to set user mode networking.

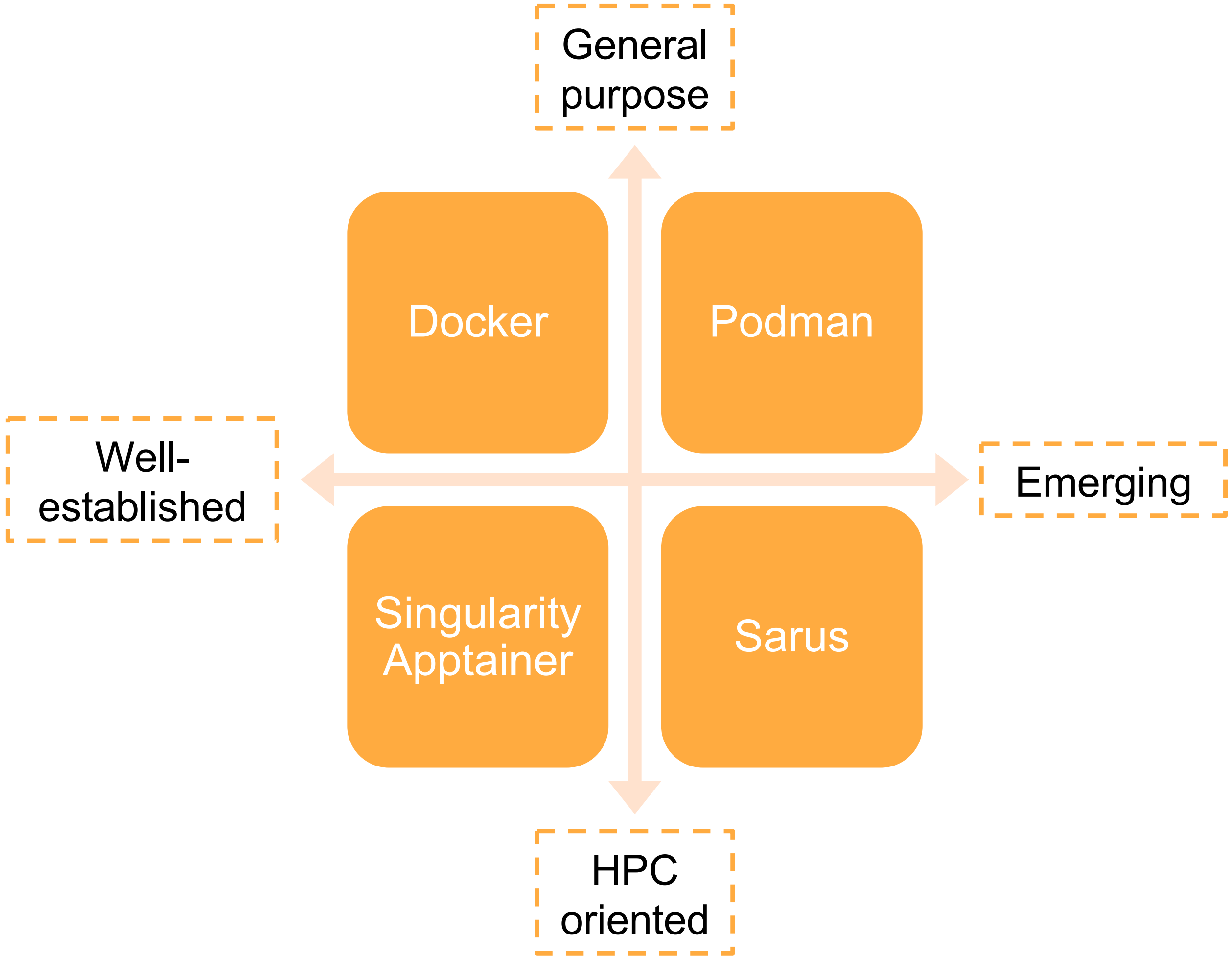
Other

- In theory, interoperable with K8s.



Choices

- Purpose: general purpose vs. HPC-oriented.
- Maturity: well-established vs. emerging
- OCI vs. proprietary formats.
- Security: risk vs. usability.



Chosen Technologies

Sarus

- OCI compliant. HPC focused.
- Best OCI solution for the short-term.

Apptainer

- Full unprivileged. HPC focused.
- Best solution for the long term, if it becomes more OCI compliant.

Podman

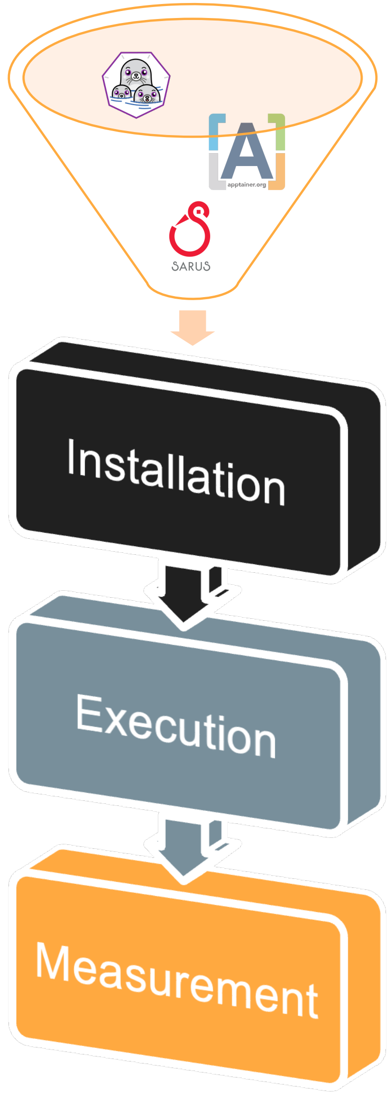
- Best docker alternative for general purpose and K8s.
- Interesting to show the limitations on HPC environments.

Deployment and Testing

- Install the technology on client cluster.
  - Tested on both RHE7 and RHE8
  - 2-nodes testing environment (node271 and node272).
    - 2x Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
  - Favor non-privileged installations with customized paths.

- Launch a containerized application using the technology.
  - HYDRO application: [cneshpc / benchs_parallel · GitLab]
  - Benchmark for testing compiler and MPI libraries installation.
  - Containerized with MPICH 3.4.3 and OpenMPI 3.1.4.

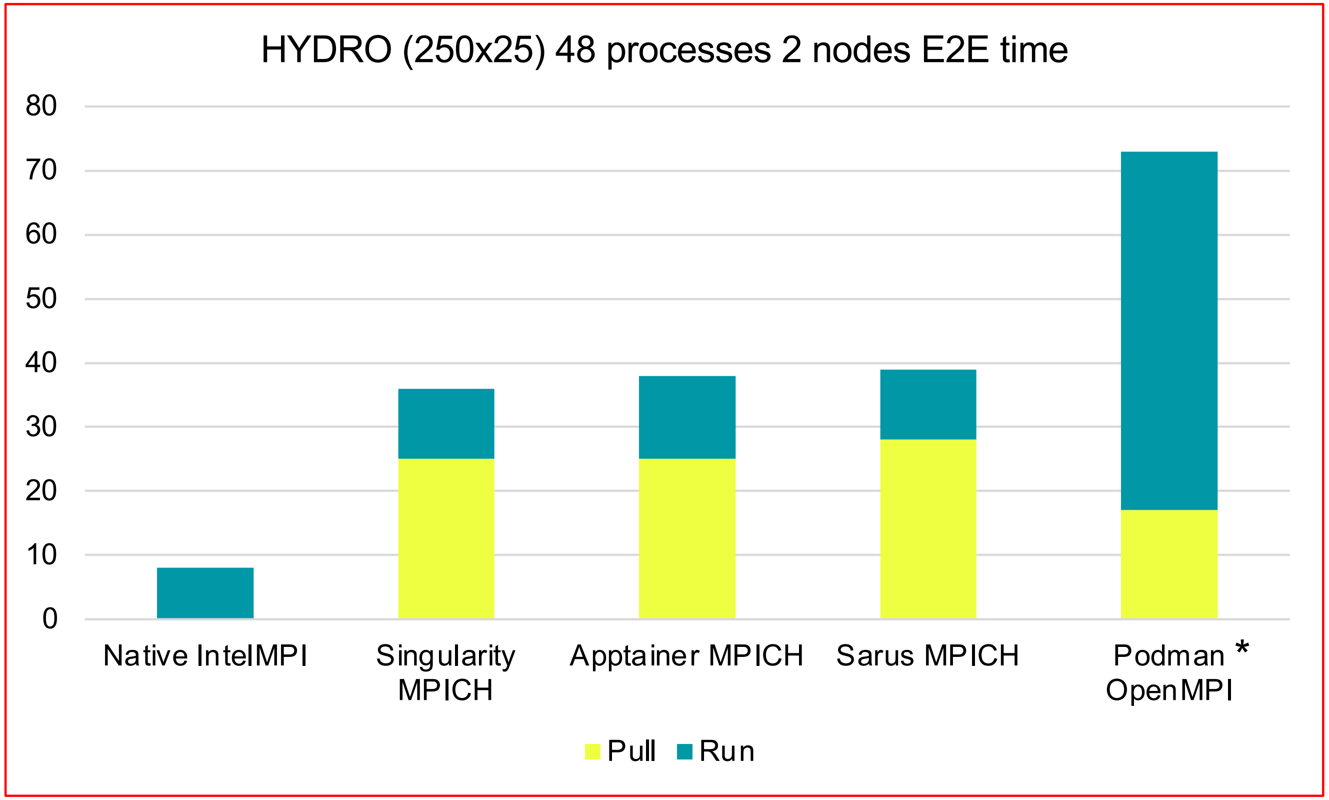
- Measure and compare performances.
  - Use singularity as baseline.



Study MPI – All Technologies

- Tests completed.
  - Pull and run times.

- Space for improvement.
  - Scratch space local to the nodes.
  - Sarus native MPI hook.
  - Podman optimization.



doitnow

Conclusions

- No single technology covers all the use cases.
- Relevance of the latest kernel features.
- Landscape continuously evolving.
- Focus on client needs. Tailored solutions.
- Next step: interactive and full-stacked container solutions for end users.

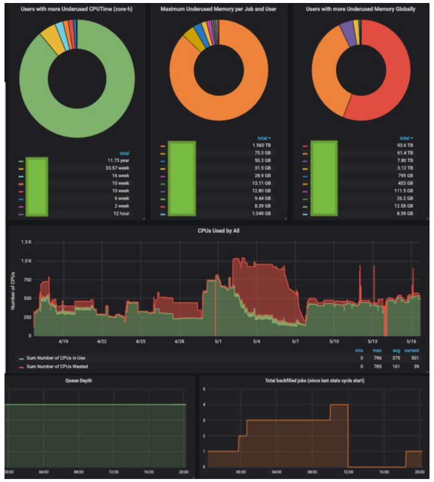
doitnow

Some references in Earth Science and Meteorogical projects

www.doit-now.tech

References 1/2

- KAUST
    - Installation of a large BeeGFS filesystem used extensively by the Earth Science Department
        - 3.8 PB, 82 GB/s, 10 data servers, 12 meta data servers
- Bureau of Meteorology (Australian Gvt)
    - Dashboards to monitor and visualize HPC clusters efficiency



References 2/2

- Bengladesh meteorological national center
  - Cluster installation in a high availability configuration
  - BeeGFS solution for storage
- Meteorological Service of Catalonia
  - Cluster installation, tunning HPC environment and applications, support.