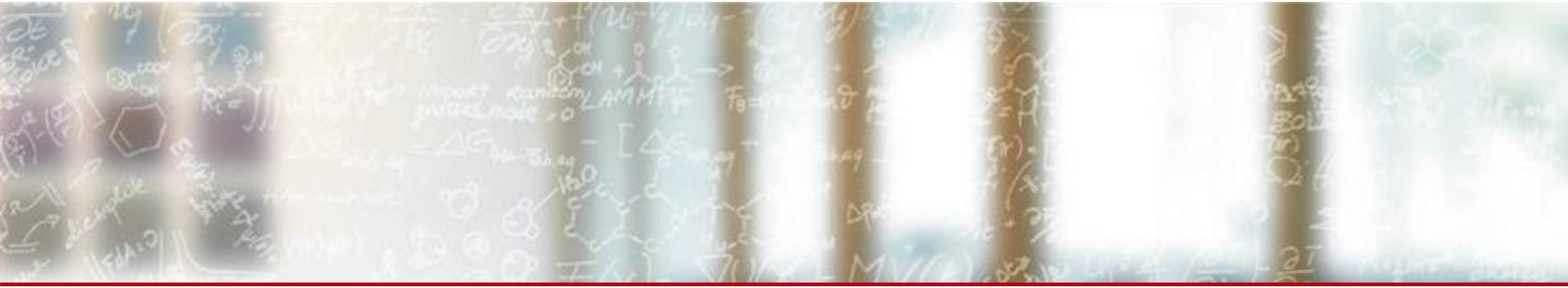
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Experiences in deploying and running Shifter

Containers for HPC, Cambridge University

Lucas Benedicic, Felipe A. Cruz, Alberto Madonna, Kean Mariotti – *Systems Integration Group*

June 30th, 2017

Outline

1. Overview
2. Docker
3. Shifter
4. Workflows
5. Use cases
6. Conclusion

Overview

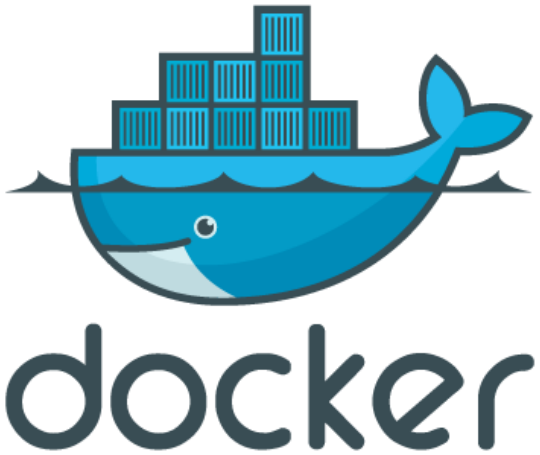
Motivation

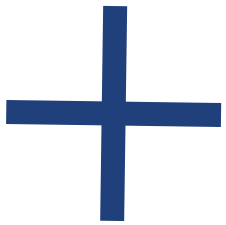
- Bring Docker containers to production on Piz Daint.
  - Docker: flexible and self-contained execution environments.
  - Tool that enable workflows for some users.
  - Part of an ecosystem that provides value to users.

- The Systems Integration group focuses on extending Shifter's container runtime.
  - Usability.
  - Robustness.
  - High performance.

In a nutshell

- Production workflows using Docker and Shifter:
  1. Build and test containers with **Docker** on a **Laptop**.
  2. Run with high-performance with **Shifter** on **Piz Daint** securely.





GPU Support

GPU Support: a user's perspective

- Singularity

```
$> module show cudatoolkit
…
setenv CRAY_CUDATOOLKIT_DIR /opt/nvidia/cudatoolkit8.0/8.0.54_2.2.8_ga620558
…
$> srun -N1 singularity --nv --bind \
/opt/nvidia/cudatoolkit8.0/8.0.54_2.2.8_ga620558 my_cuda_image.img cudaApp
```

GPU Support: a user's perspective

- Singularity

```
$> module show cudatoolkit
…
setenv CRAY_CUDATOOLKIT_DIR /opt/nvidia/cudatoolkit8.0/8.0.54_2.2.8_ga620558
…
$> srun -N1 singularity --nv --bind
/opt/nvidia/cudatoolkit8.0/8.0.54_2.2.8_ga620558 my_cuda_image.img cudaApp
```

- Shifter

```
$> srun -N1 shifter -image=my_cuda_image cudaApp
```

GPU Support: a user's perspective

- Singularity

```
$> module show cudatoolkit
…
setenv CRAY_CUDATOOLKIT_DIR /opt/nvidia/cudatoolkit8.0/8.0.54_2.2.8_ga620558
…
$> srun -N1 singularity --nv --bind /opt/nvidia/cudatoolkit8.0/8.0.54_2.2.8_ga620558
my_cuda_image.img cudaApp
```

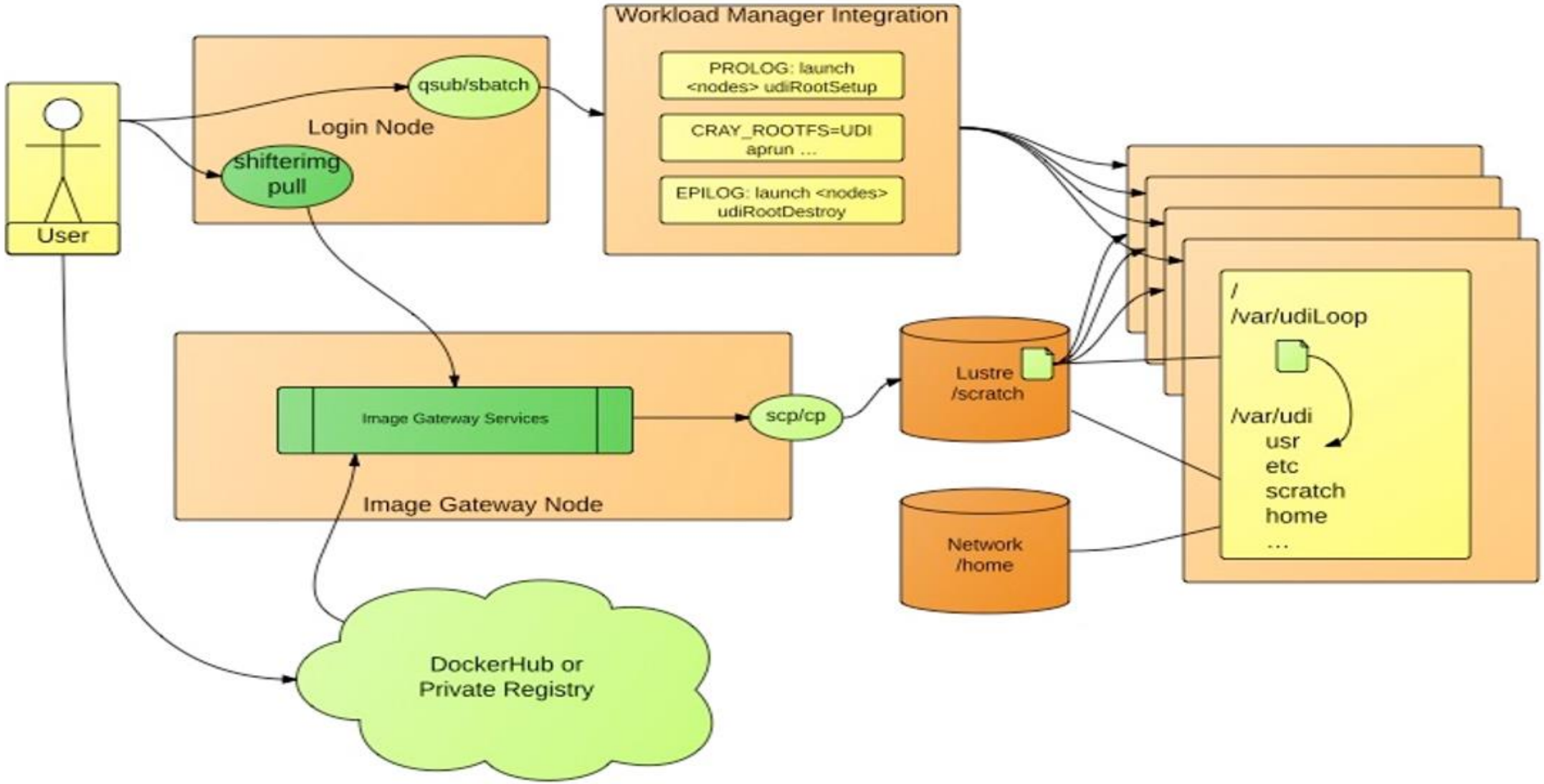
- Shifter

```
$> srun -N1 shifter -image=my_cuda_image cudaApp
```

```
root@daint> cat udiRoot.conf
…
siteResources=/opt/shifter/site-resources/cuda:/opt/shifter/site-resources/nvidia
…
```

Shifter Internals

Shifter Internals



Use case: TensorFlow (GPU / Third party container)

TensorFlow

- Software library capable of building and training neural networks using CUDA.
- **Official** TensorFlow image from DockerHub (not modified).
- TensorFlow has a **rapid release cycle** (Once a week new build available!).
- **Ready to run** containers.
- Performance relative to the Laptop wall-clock time of image classification tests.

Test case	Laptop*	Piz Daint (P100)
MNIST, TF tutorial	**613** [seconds]	**17.17**x

*Laptop run using **nvidia-docker**

Use case: Large Hadron Collider

Atlas and LHC

- CSCS operates a cluster running experiments of the LHC at CERN
- Jobs expect a RHEL-compatible OS and precompiled software stack
- Shifter reproduces the certified software stach on Piz Daint (Cray XC50)

Use case: OSU benchmark (MPI)

OSU Benchmark

```
$ srun -n2 -N2 shifter --mpi --image=osu-benchmarks-image ./osu_latency
```

- Host MPI:
  - Cray MPT 7.5.0
  - Cray Aries Interconnect
- Container MPI:
  - MPICH v3.1 (A)
  - MVAPICH2 2.2 (B)
  - Intel MPI Library (C)
- Native performance!

		Shifter MPI support *Enabled*			Shifter MPI support *Disabled*		
Size	Native	A	B	C	A	B	C
32	1.1	1.00	1.00	1.00	4.35	6.17	4.41
128	1.1	1.00	1.00	1.00	4.36	6.15	4.51
512	1.1	1.00	1.00	1.00	4.47	6.22	4.56
2K	1.6	1.06	1.00	1.06	4.66	5.03	4.04
8K	4.1	1.00	1.02	1.02	2.17	2.02	1.86
32K	6.5	1.03	1.03	1.03	2.10	2.17	1.91
128K	16.4	1.01	1.01	1.01	2.63	2.84	1.95
512K	56.1	1.00	1.01	1.01	2.23	1.78	1.67
2M	215.7	1.00	1.00	1.00	2.02	1.41	1.37

Table 4: Results from `OSU_latency` on Piz Daint: Native runs use Cray MPT 7.5.0 over Cray Aries interconnect; relative performance against native is reported for containers with (A) MPICH 3.1.4, (B) MVAPICH2 2.2, and (C) Intel MPI library using Shifter with MPI support *enabled* and *disabled*.

Use case: PyFR (GPU + MPI)

PyFR

- **Python** based framework for solving advection-diffusion type problems on streaming architectures. 2016 **Gordon Bell** Prize finalist (Highly scalable).
- **GPU-** and **MPI-accelerated** runs using containers.
- Complex build (100 lines Dockerfile) and test on Laptop.
- Production-like run on Piz Daint.
- Parallel efficiency for a 10-GB test case on different systems (4 node setup).

Number of nodes	Piz Daint (P100)
1	1.000
2	0.975
4	0.964
8	0.927
16	0.874

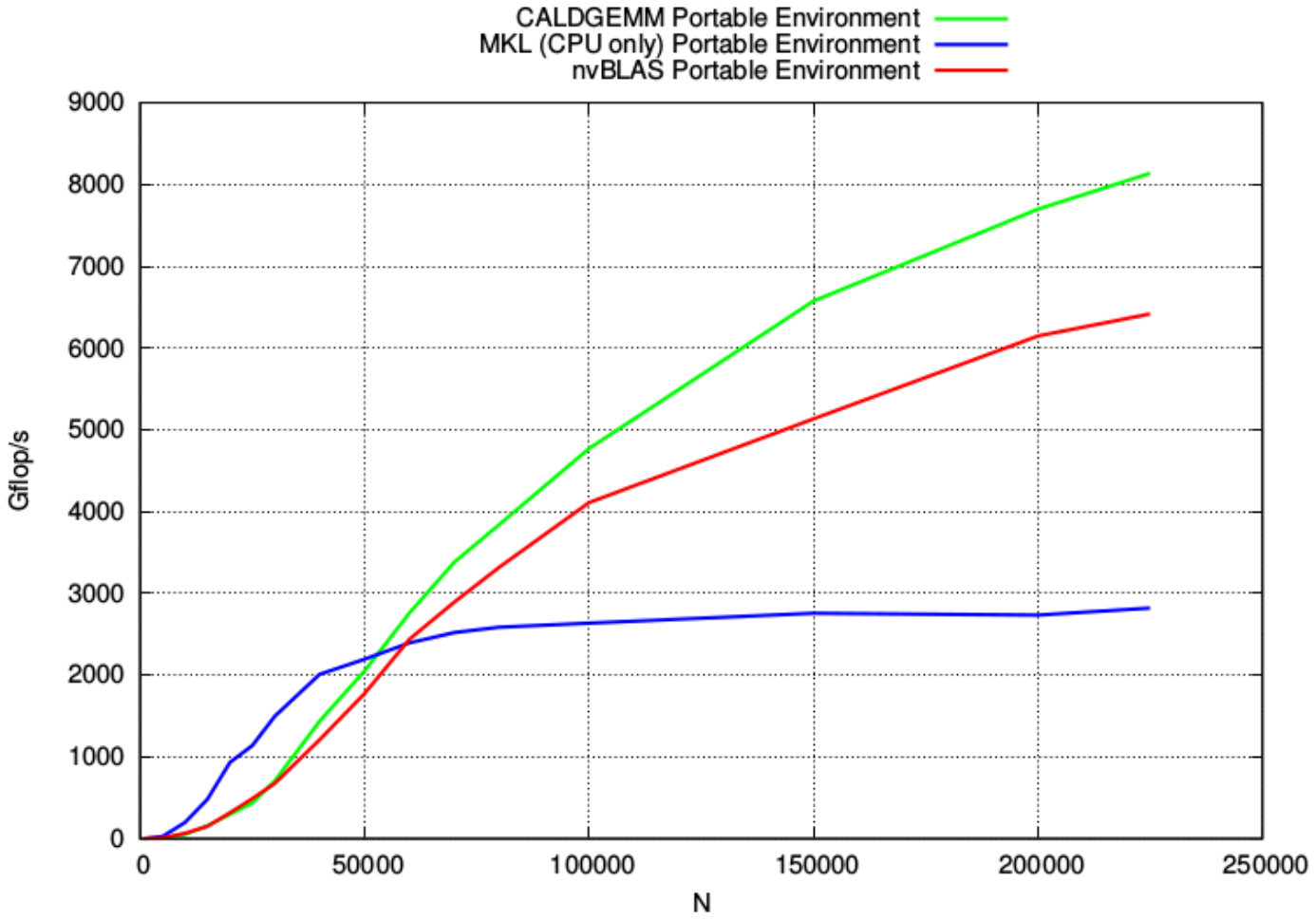
Use case: Portable compilation units

Vanilla Linpack with specialized BLAS

- Some application performance depends on targeted optimization of libraries.
- Use container to pack application environment.
- Two stage: compile first (link against host libs), then run.

Vanilla Linpack with specialized BLAS

- Some application performance depends on targeted optimization of libraries.
- Use container to pack application environment.
- Two stage: compile first (link against host libs), then run.
- Proof of concept: pack vanilla Linpack, compile specialized BLAS before run.



Conclusion

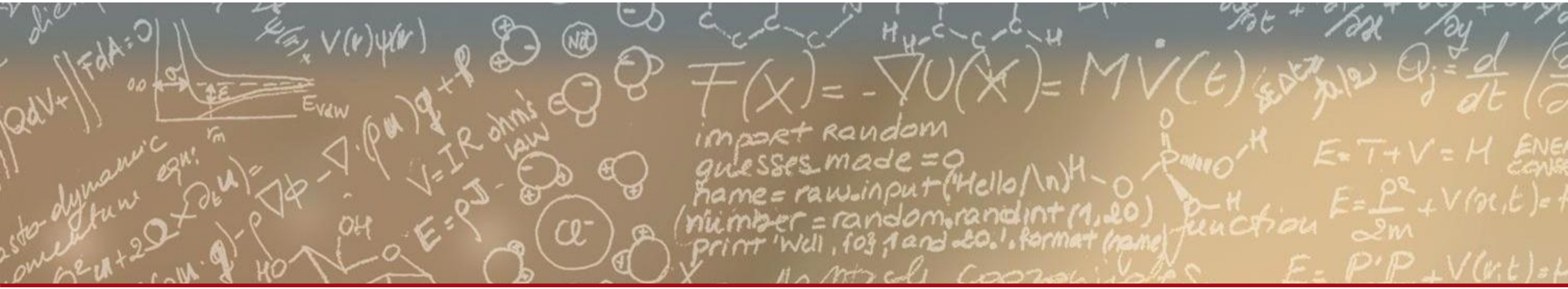
Conclusion

- The showed use cases highlighted:
  - pull and run containers;
  - high-performance containers;
  - access to hardware accelerators like GPUs;
  - use of high-speed interconnect through MPI;
  - portable compilation environments.

Conclusion

- The showed use cases highlighted:
  - pull and run containers;
  - high-performance containers;
  - access to hardware accelerators like GPUs;
  - use of high-speed interconnect through MPI;
  - portable compilation environments.

- Linux container technology is here to stay
  - >95% of the nice container features are available on **all** implementations
  - REMEMBER: the decision about which technology to choose should be driven by the workflows within your organization!



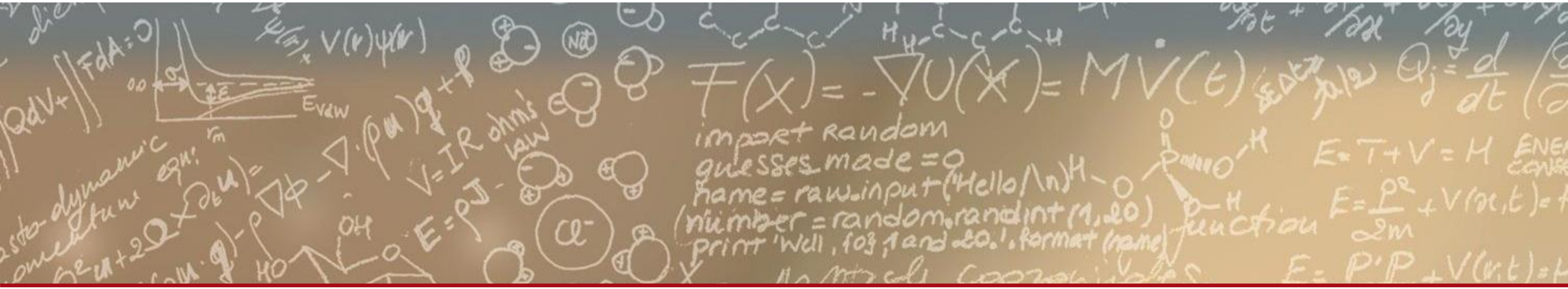
Soon to be announced …

OAK RIDGE National Laboratory

Tokyo Institute of Technology ATTUNE

NERSC
National Energy Research Scientific Computing Center

CRAY
THE SUPERCOMPUTER COMPANY



Thank you for your attention