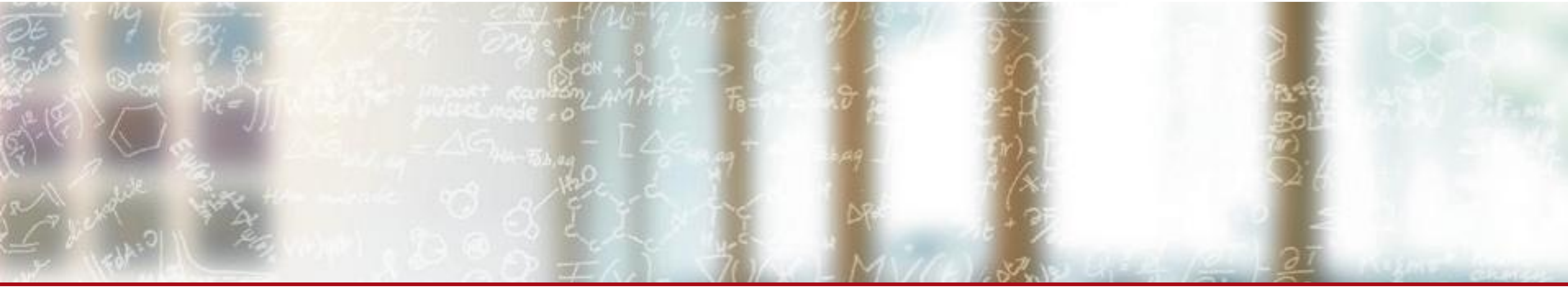
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Introduction to the Sarus container engine

PRACE HPC Workshop on Containers and Unikernels

Lucas Benedicic, ETH Zurich / CSCS
Alberto Madonna, ETH Zurich / CSCS

July 7, 2021

Table of Contents: morning

1. 9:00 - 9:20 Introduction to Sarus form a user’s perspective
   9:20 - 9:30 Q&A

2. 9:30 - 9:45 Installing Sarus on your system
   9:45 - 10:15 Hands on: installing Sarus on a Debian 10 VM

10:15 - 10:45 Break

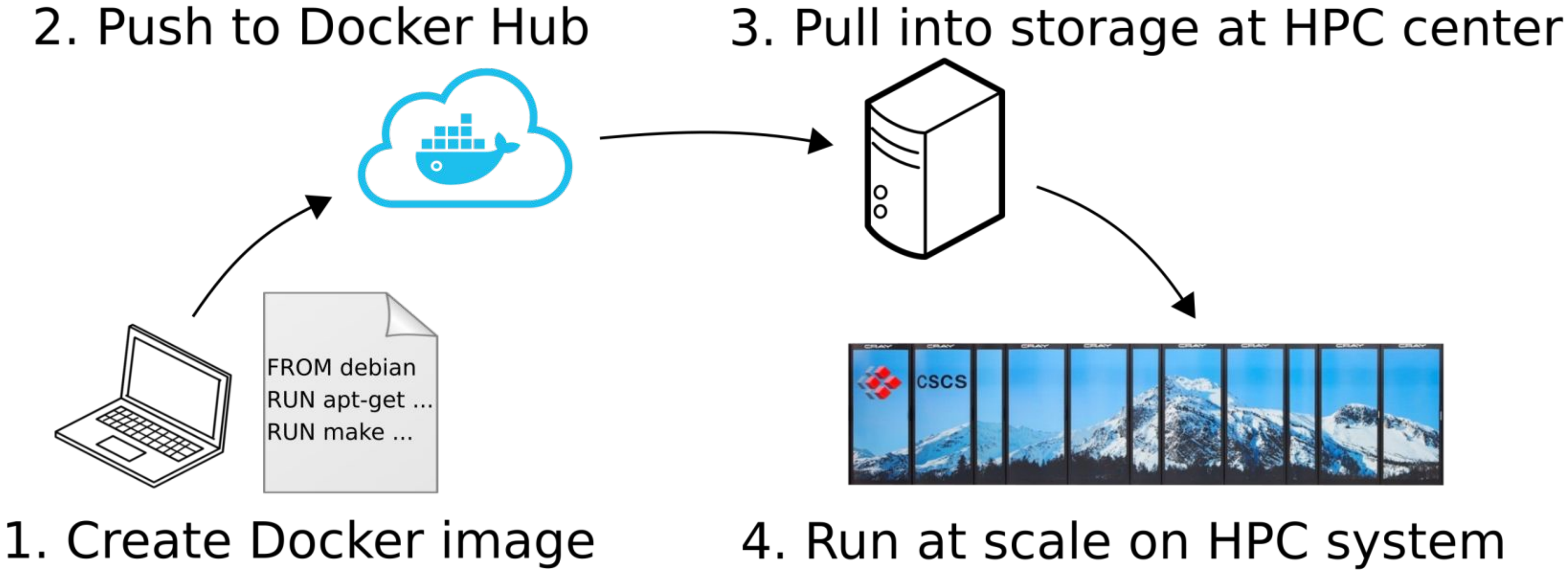
3. 10:45 - 11:15 MPI examples with OSU benchmarks
   11:15 - 12:00 Hands on:
   - Sarus basic commands
   - MPI exercises

12:00 - 13:00 Lunch break

Sarus container engine

- Combines container portability with native HPC performance
- Integrates with HPC infrastructure and software
- Customizes containers at runtime with standard plugins
- Pulls regular Docker images
- Provides a Docker-like command line interface

Typical user workflow at CSCS



2. Push to Docker Hub

3. Pull into storage at HPC center

```
FROM debian
RUN apt-get ...
RUN make ...
```

1. Create Docker image

4. Run at scale on HPC system

Highlights of Sarus from a user perspective

- Consistent experience
  - With Docker: closely resembling CLI
  - With host environment: env variables, uid/gid, file permissions
- Pull images from Docker registries (e.g. Docker Hub, NVIDIA NGC)
- Import images from local tar archives (no cloud upload required)
- Integration with the workload manager (Slurm)
- Native performance from GPUs and high-speed interconnects
- Access to parallel filesystems inside containers

Sarus CLI

- Sarus

```
# pull image
$ sarus pull [options] <image>[<:tag>]

# load image
$ sarus load [options] <file> <image>

# show list of images
$ sarus images

# remove image
$ sarus rmi <image>[<:tag>]

# run container
$ sarus run [options] <image>[<:tag>]
<command> <args>
```

- Docker

```
# pull image
$ docker pull [options] <image>[<:tag>]

# load image
$ docker load [options] -i <file>

# show list of images
$ docker images [options] [repo[<:tag>]]

# remove image
$ docker rmi [options] <image> [image...]

# run container
$ docker run [options] <image>[<:tag>]
<command> <args>
```

Further reading

- Sarus on the CSCS User Portal:
  https://user.cscs.ch/tools/containers/sarus/

- Code on GitHub:
  https://github.com/eth-cscs/sarus

- User documentation on Read the Docs:
  https://sarus.readthedocs.io/en/stable/user/index.html

- Benedicic, L., Cruz, F.A., Madonna, A. and Mariotti, K., 2019, June. Sarus: Highly Scalable Docker Containers for HPC Systems. In *International Conference on High Performance Computing* (pp. 46-60). Springer, Cham.
  https://doi.org/10.1007/978-3-030-34356-9_5

Installation tutorial

Hands on!

EURO

Break time! Back at 10:45

MPI containers on Piz Daint

EURO

MPI containers on Piz Daint

- Generic images can run unmodified by instructing Slurm to use the PMI-2 interface:

```
srun --mpi=pmi2 sarus run <image> <args>
```

- This way, containers will use the MPI libraries from the image and run at sub-optimal performance

- Images using MPICH and derivatives: work out of the box

- Images using OpenMPI: OpenMPI must be built with PMI-2 support
  - Configure example on Ubuntu 18.04:

```
./configure --prefix=/usr --with-pmi=/usr/include/slurm-wlm --with-pmi-libdir=/usr/lib/x86_64-linux-gnu \
 CFLAGS=-I/usr/include/slurm-wlm
```

MPI containers on Piz Daint

- Images using MPICH-based implementations can take advantage of ABI compatibility (https://www.mpich.org/abi/)

- Sarus can replace the image MPI with host libraries at runtime, achieving the full performance of the Cray Aries interconnect:

```
srun sarus run --mpi <image> <args>
```

- Recommended libraries for compatibility with Piz Daint:

MPICH 3.1.4
MVAPICH2 2.2
Intel MPI Library 2017 Update 1

Hands on!

EURO

EURO

Lunch break! Back at 13:00

Table of Contents: afternoon

4. 13:00 - 13:30 GPU examples with CUDA SDK
   13:30 - 14:30 Hands on: CUDA samples or user-provided GPU applications

14:30 - 15:00 Break

5. 15:00 - 15:30 Real-world application with data I/O: GROMACS
   15:30 - 16:30 Hands on with GROMACS or other applications

6. 16:30 - 17:00 Q&A with the trainers

17:00 End of day

GPU containers on Piz Daint

GPU containers on Piz Daint

- When running on Piz Daint's GPU nodes, GPU devices are automatically added to containers

- Fastest way to get CUDA in a Dockerfile: use NVIDIA official images! https://hub.docker.com/r/nvidia/cuda

```
FROM nvidia/cuda:11.3.0-devel-ubuntu20.04
```

- NVIDIA images are provided for Ubuntu, Red Hat UBI and CentOS
  - Other distributions can still install the CUDA Toolkit through package manager or runfile

- The NVIDIA driver should NOT be installed in the image (it's bound to the hardware!)

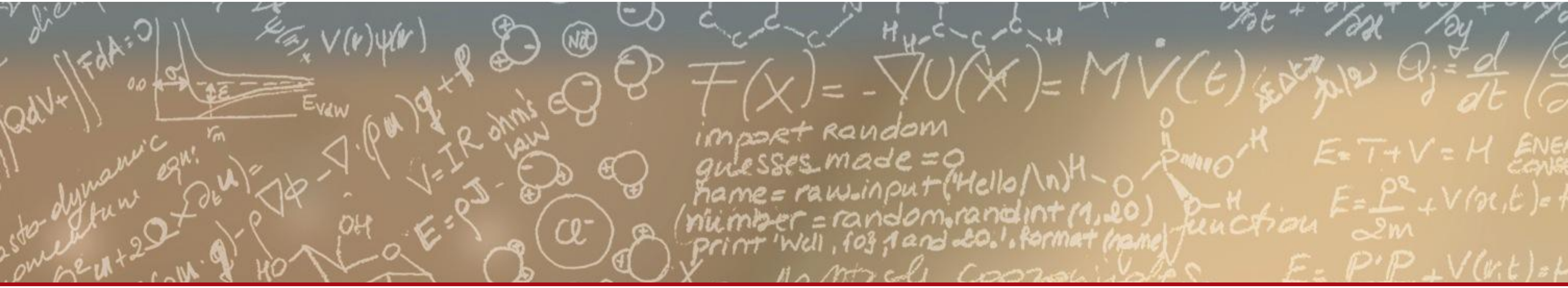
Hands on!

Break time! Back at 15:00

EURO

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your attention.

EURO

Backup slides

Docker and HPC: not a good fit

- Security model assumes root privileges
- No integration with workload managers
- Missing support for diskless nodes
- Very limited support for kernel bypassing devices (e.g. accelerators and NICs)
- No adequate parallel storage driver