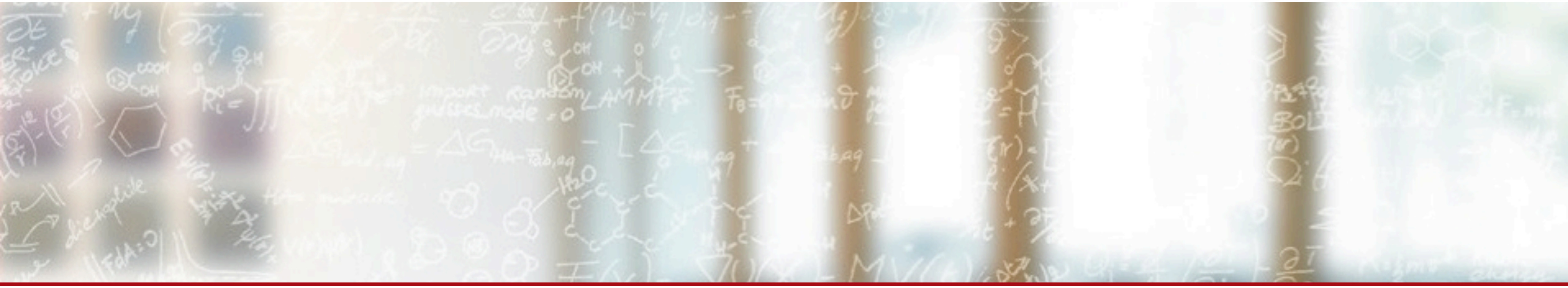
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Docker and Shifter: portable and performant scientific computing

SOS 21 Workshop, Davos, Switzerland

Lucas Benedicic, CSCS
March 21st, 2017

Outline

1. Background
2. HPC Use Cases
3. Data Science Use Cases
4. Conclusion

Background

About Docker

- Process Container
  - It uses Linux kernel features to create semi-isolated “containers”.
  - Captures all application requirements.
- Image Management
  - Easy-to-use recipe file.
  - Version-control driven image creation.
- Environment
  - Pull and Pull images from a community-driven Hub (i.e., DockerHub)

docker

About Shifter (1)

- A runtime to increase flexibility and usability of HPC systems by enabling the deployment of Docker-like Linux containers.
- Originally developed at NERSC by D. Jacobsen and S. Canon.
- **Flexibility**
  - Enable the definition of complex software stacks using different Linux flavors.
  - Develop an application on your laptop and run it on an HPC system.
- **Integration**
  - Availability of shared resources (e.g., parallel file systems, accelerator devices and network interfaces).
- **Compatibility**
  - Integration with public image repositories, e.g., DockerHub.
  - Improving result reproducibility.

About Shifter (2)

- But containers are hardware- and platform-agnostic by design
  - How do we go about accessing specialized hardware like GPUs?
- CSCS and NVIDIA co-designed a solution that provides:
  - direct access to the GPU device characters;
  - automatic discovery of the required libraries at runtime;
  - NVIDIA's DGX-1 software stack is based on this solution.
- CSCS extended this design to the MPI stack.
  - Supports different versions MPICH-based implementations.

HPC Use Cases

N-body simulation (1)

- Let’s start with Docker on the laptop

```
$ nvidia-docker pull nvidia/cuda:8.0-devel-ubuntu14.04
8.0-devel-ubuntu14.04: Pulling from nvidia/cuda

$ nvidia-docker run nvidia/cuda:8.0-devel-ubuntu14.04 \
  nbody -benchmark -device=0 -numbodies=2000000 -fp64
```

- Let’s now move to an HPC system with Shifter

```
$ shifterimg pull docker:nvidia/cuda:8.0-devel-ubuntu14.04
Pulling from nvidia/cuda …

$ srun shifter --image=nvidia/cuda:8.0-devel-ubuntu14.04 \
  nbody -benchmark -device=0 -numbodies=2000000 -fp64
```

N-body simulation (2)

- Successful GPU-accelerated runs using the official CUDA image from DockerHub.
- GFLOP/s performance of a double-precision, 200k-body simulation on different systems.

	Laptop*	GPU cluster (K40)	Multi-GPU cluster (K40-K80)	Piz Daint (P100)
Native	18.34	858.09	1895.32	2733.01
Shifter	18.34	858.48	1895.17	2733.42

*Laptop run using **nvidia-docker**

PyFR

- Python based framework for solving advection-diffusion type problems on streaming architectures.
- 2016 Gordon Bell Prize finalist.
- Successful GPU- and MPI-accelerated runs using the same container image.
- Parallel efficiency for a 10-GB test case on different systems.

Number of nodes	Laptop	GPU cluster (K40)	Piz Daint (P100)
1	-	1.000	1.000
2	-	0.987	0.975
4	-	-	0.964
8	-	-	0.927
16	-	-	0.874

Trilinos (1)

- A collection of open-source software libraries, intended to be used as building blocks for the development of scientific applications.
- Several supercomputing facilities provide a native version of Trilinos for their users.
- Sean Deal, author of *HPC Made Easy: Using Docker to Test and Distribute Trilinos,* published a Docker container featuring Epetra with MPI support.

Trilinos (2)

- Replaced the OpenMPI library in the container with vanilla MPICH.
- Test problem: 12 MPI processes, 1000x1000 mesh nodes and a 25 points stencil.
- Successful MPI run on a laptop (Docker)

```
$ docker run ethcscs/trilinos-epetrampi-benchmark mpirun -n 12 \
  Epetra_SjdealBenchmark.exe 1000 1000 3 4 25 -v
Epetra::MpiCommEpetra::MpiCommEpetra::MpiCommEpetra::MpiCommEpetra in Trilinos
12.10.1
```

- Successful MPI run on Cray XC50 (Shifter)

```
$ srun -n12 -N1 shifter --mpi --image=ethcscs/trilinos-epetrampi-benchmark \
  Epetra_SjdealBenchmark.exe 1000 1000 3 4 25 -v
Epetra::MpiCommEpetra::MpiCommEpetra::MpiCommEpetra::MpiCommEpetra in Trilinos
12.10.1
```

Data Science Use Cases

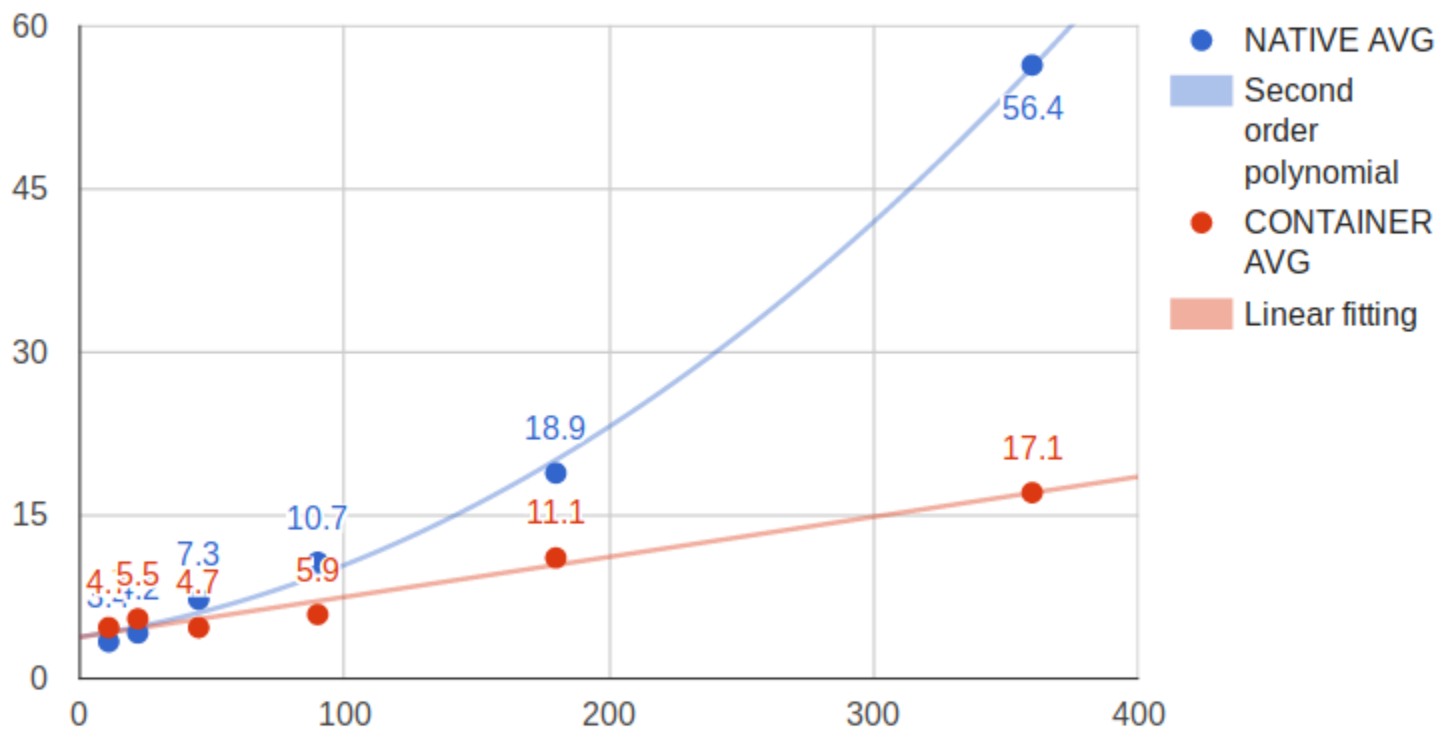
TensorFlow (1)

- Software library capable of building and training neural networks to detect and decipher patterns and correlations.
- Successful GPU-accelerated runs using the official TensorFlow image on DockerHub.
- Wall-clock times for two test cases on different systems.

Test case	Laptop	GPU cluster (K40)	Piz Daint (P100)
MNIST, TF tutorial	613.24	104.92	35.74
CIFAR-10, 100k iterations	23359.00	8905.00	6246.00

Apache Spark

- Designed around commodity clusters, i.e., Ethernet and local disks.
- Does not scale well on parallel filesystems.
- Shifter minimizes the file-system metadata overhead.



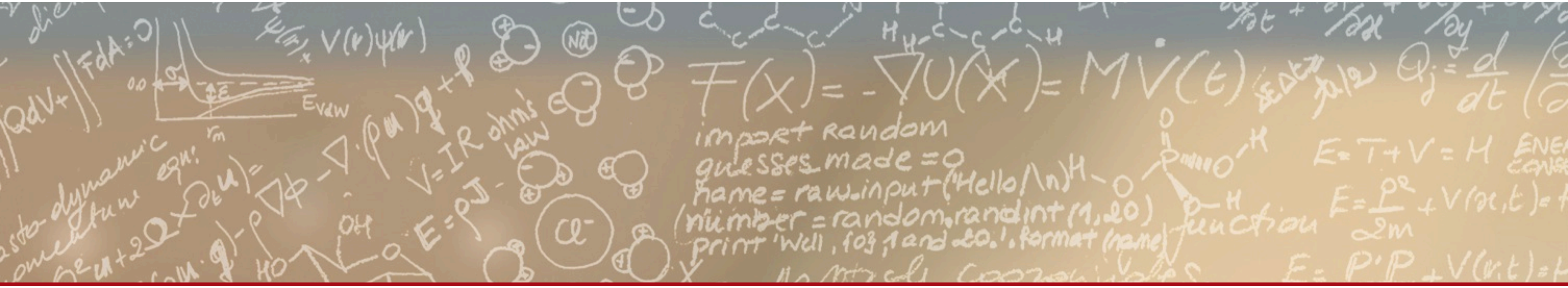
Conclusion

Conclusion

- Containers are here to stay.
- The Docker-Shifter combo takes us closer to the turn-key, cloud-based like computing with scalability and high-performance.
- The showed use cases highlighted:
  - bare metal provisioning;
  - ready to use, high-performance software stacks;
  - network file systems support;
  - access to hardware accelerators like GPUs and high-speed interconnect through MPI.

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for you attention