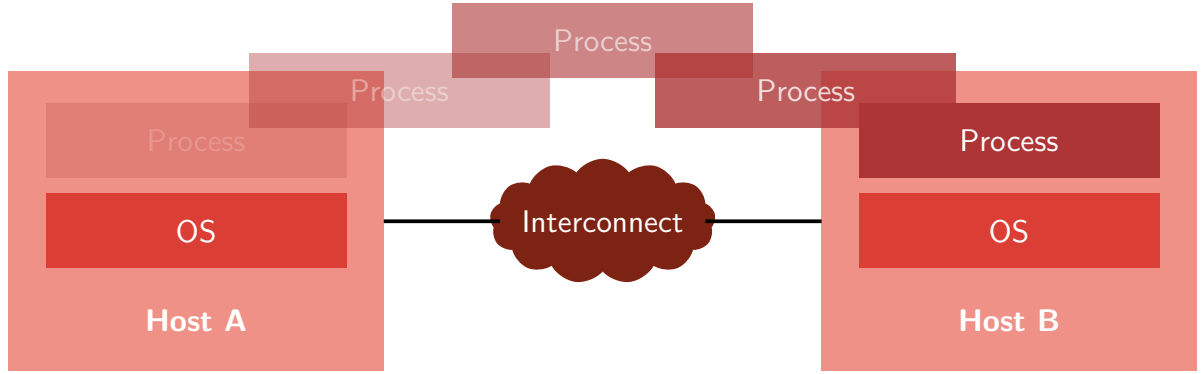


Migrating LinuX Containers Using CRIU

Simon Pickartz, Niklas Eiling, Stefan Lankes,
***Lukas Razik*, and Antonello Monti**

RWTH Aachen University, Aachen, Germany

Why do we need Migration?

Resiliency

- **Increasing hard- and software failures** with **growing** cluster sizes
- Evacuation of **faulty nodes** $\Rightarrow$ **no whole job aborts**

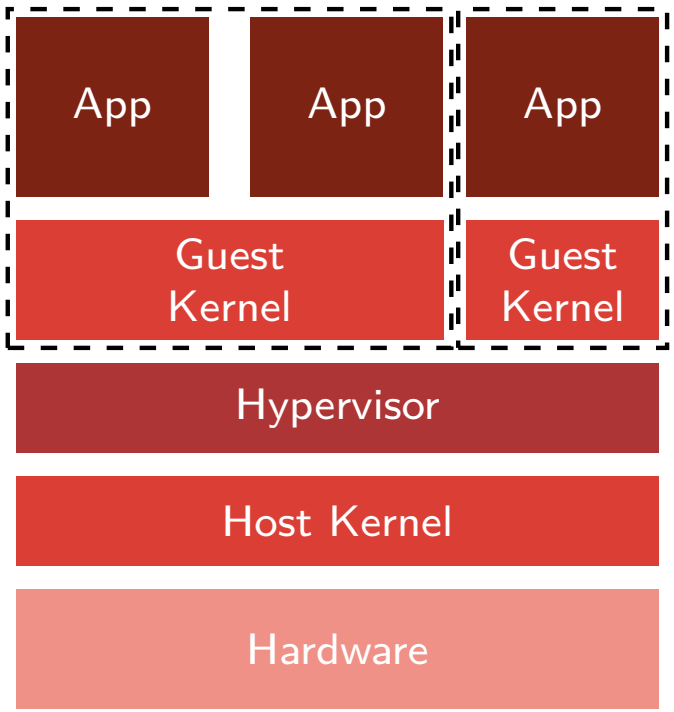
Load balancing

- Applications' **scalability** is usually **limited by a single resource**
- **Co-scheduling** can **improve the overall cluster utilization**
  - **Revocation** of an exclusive **node assignment**
  - **Dynamic schedules** necessary

Why should we care about containers?

Virtual Maschines

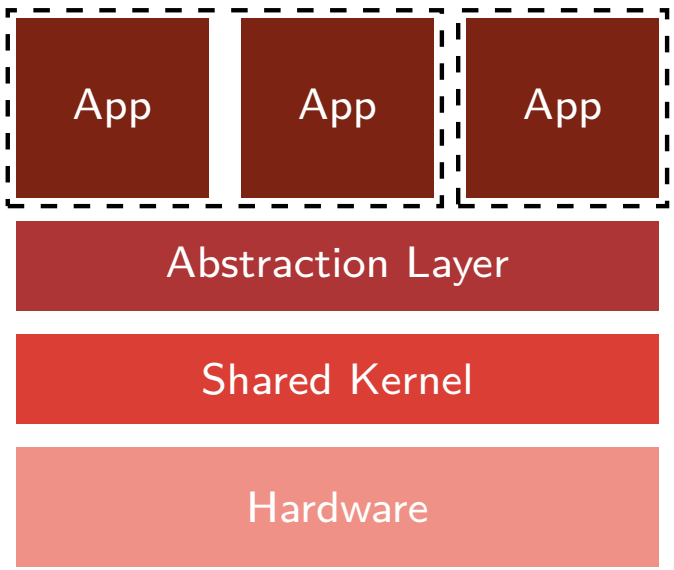
- Virtualization *of hardware*
- Multiple kernels
- Device emulation
- **Requires special hardware support for nearly native performance**
- **Flexibel** in terms of kernel / OS choice



Why should we care about containers?

Containers

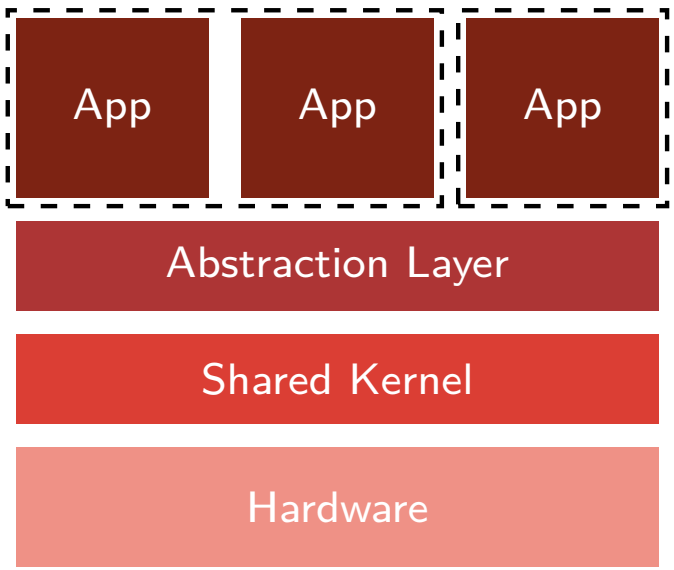
- **Re-use host kernel**
- **User-space instances** defined and **separated** by so-called
  - *cgroups*
  - *namespaces*
- **No special hardware support required**
- **Restricts** kernel / OS choice



Why should we care about containers?

Containers

- **Re-use host kernel**
- **User-space instances** defined and **separated** by so-called
  - *cgroups*
  - *namespaces*
- **No special hardware support required**
- **Restricts** kernel / OS choice



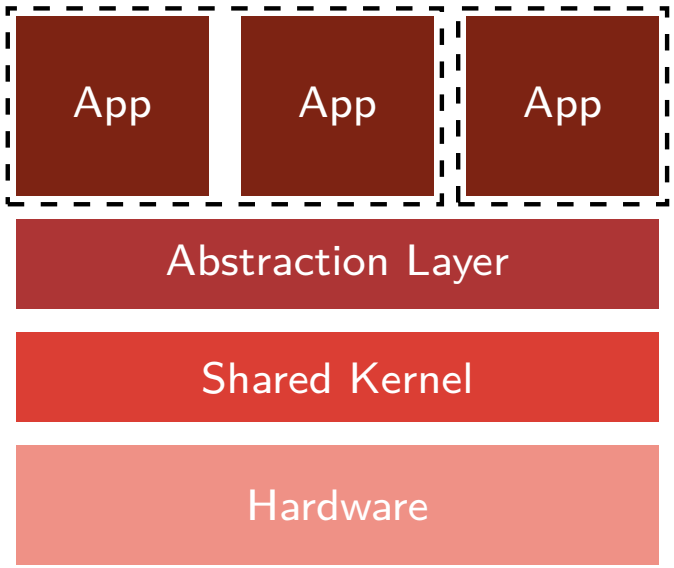
⇒ **Containers are *fast* and light-weight**

Agenda

- Background
  - Container-based Virtualization in terms of HPC
  - Libvirt
  - Checkpoint / Restore in Userspace (CRIU)
- Lxctools Driver
  - Integration into Libvirt
  - Migration Based on CRIU
- Evaluation
  - lxctools Driver Overhead
  - Migration Time
- Conclusion

Container-based Virtualization in terms of HPC

- **Isolation at OS level** which is **lower than** with **hypervisors**
  - ⇒ Container may crash whole system
- **No overhead** introduced by **multiple kernels**
- **Flexible resource allocation**
- Potentially **better I / O and CPU performance** than **common hypervisor-based** approaches
  - ⇒ Complies with the **goals of HPC**



Namespaces and Cgroups

- Namespaces and Cgroups are **Linux Vanilla Kernel** features
- **Namespaces** separate **processes** by **groups**
- **Cgroups** control, limit, and observe the **resource** usage **of processes**

Namespaces

pid	Process IDs
net	Network configuration
ipc	Inter-process communication
mnt	Mountpoints
uts	Hostname
user	User and group IDs

cgroup subsystems

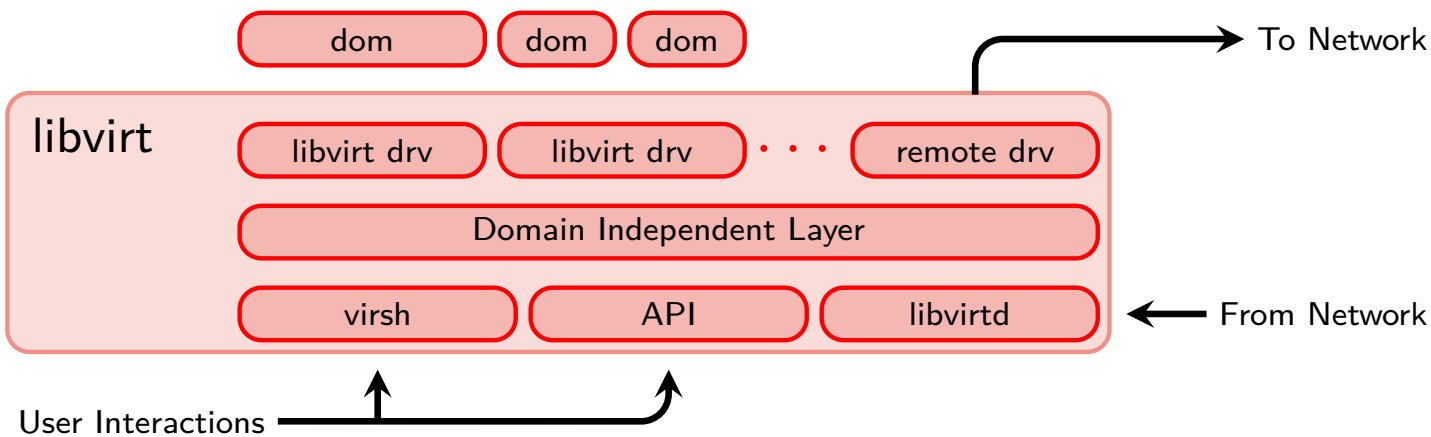
cpu	CPU time and utilization
cpuset	Amount of CPU cores
blkio	Limit / observe I / O accesses
memory	Limit memory consumption / utilization

LinuX Containers

- **Infrastructure** for container projects (e. g. **Docker**)
- ***Linux Conainers* (LXC)** use **cgroups** and **namespaces**
- **User-space solution** $\Rightarrow$ needs **no customized kernels**
- Ships with
  - **C-API**
  - Command-Line Interface (**CLI**)

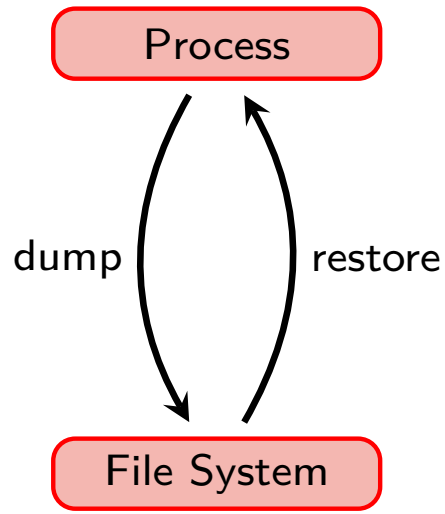
Libvirt

- **Domain (VM etc.) Management library** written in C
- **Support** for a variety of virtualization solutions,
  e. g. **QEMU**, **VMware ESX**, **OpenVZ**
- **Domain Independent Layer**
  - ⇒ **Independence of changes to virtualization layer**
  - ≡ **Implements common functionality** (e. g., configuration via XML)
- **Domain Dependent Layer**
  - ≡ Implemented in terms of ***drivers***
  - ≡ **Remote driver** enabling remote management of domains



Checkpoint / Restore in Userspace (CRIU)

- **Checkpoint / Restore (C / R) without kernel modifications**
- C / R **processes or groups** thereof
- Supports **incremental checkpoints**
- Uses `ptrace` to **freeze a process** and **inject *parasite code***
  - Gathers **file descriptors, memory maps, and register** contents
  - Requires **root privileges**
- Provides direct **memory transfer** to a **page-server** running **on remote nodes**

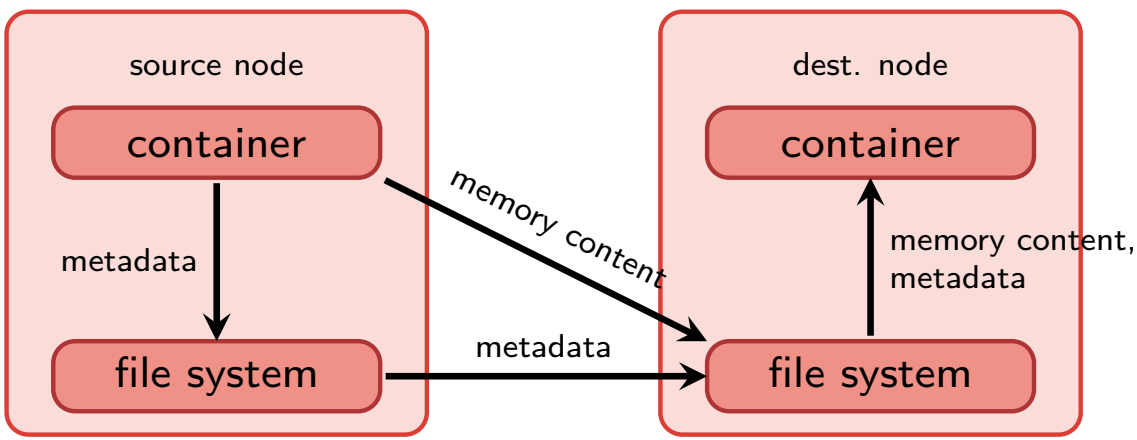


The lxctools Driver

- Utilizes **LXC** and **CRIU** for usage by libvirt
- **Minor libvirt** source code **modifications**
  - Make the **driver public to libvirt** (7 source files)
  - **Link** against the **LXC's C-API**
- **Remote management** capabilities by **libvirt**
- **Implemented features**
  - **Start / Stop** containers
  - **Checkpoint / Restore** containers
  - **Migrate** containers (**cold- and live**-migration)
  - **Configuration** of LXC instances **via XML**
- **lxctools** driver **not related** to libvirt's **lxc** driver
  libvirt's **lxc** driver provides **no migration** support

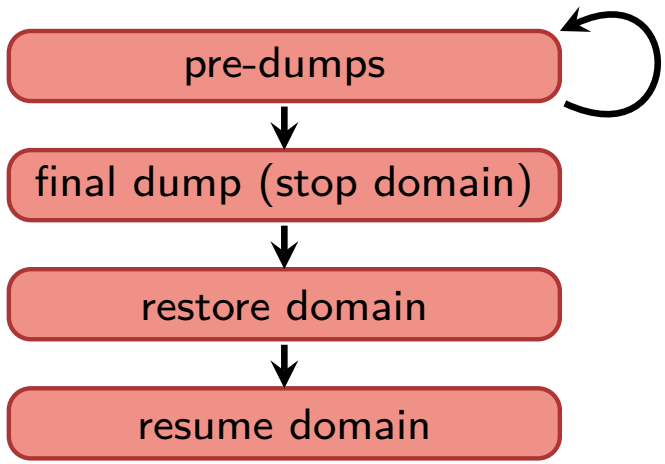
Migration by lxctools Driver

- **Implements skeleton** of libvirt's **domain-independent** layer
- **Reduce file system overhead** by using `tmpfs`
- **Migration** in four steps
  1. **Create** `tmpfs` on **both** nodes
  2. **Start page-server** on **target** node
  3. **Dump process** and
     - = **transfer memory** content via **page-server**
     - = **transfer other checkpointing data** via **file-transfer**
  4. **Restore process** from dump **on target system** and remove `tmpfs`



Live-Migration by lxctools Driver

- **Live-migration** leverages **incremental dumps**
- **Static** and **dynamic iteration counts** are **supported**
- **Pre-dumps** only for **memory content**
- Common **pre-copy live-migration** approach



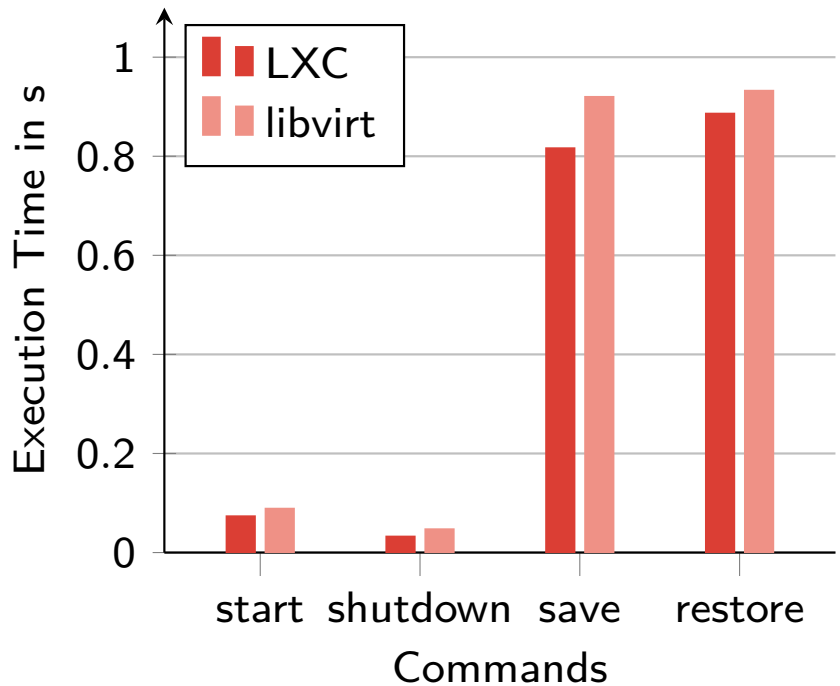
lxctools Driver Overhead

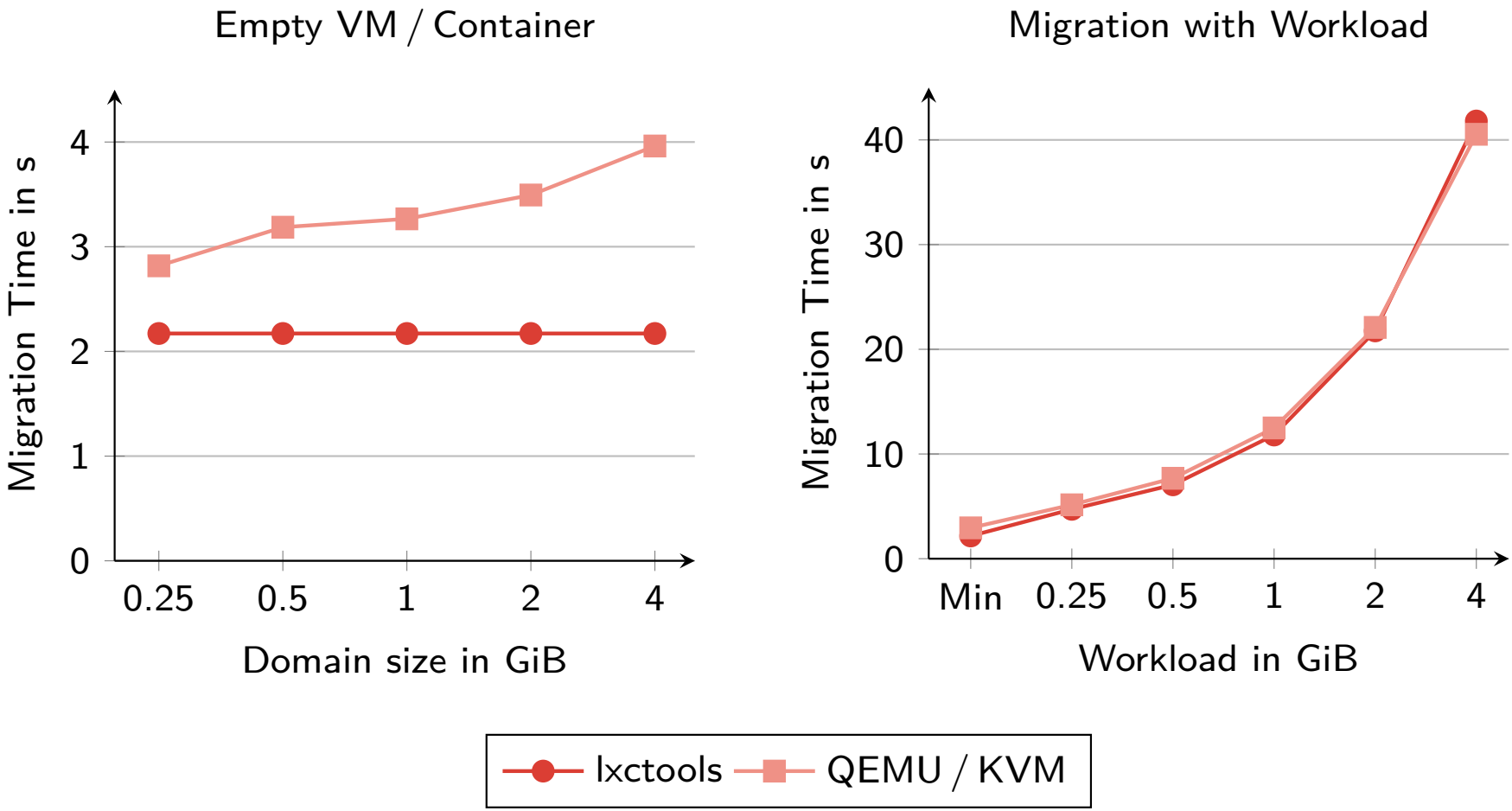
Systems:

- 2 NUMA nodes, each with 2 sockets
- Intel IvyBridge CPUs (E5-2650 v2), 8 cores (16 threads) @2.6 GHz
- Gigabit Ethernet
- Fedora 23, Linux Kernel v4.4.4-301, libvirt v1.2.16

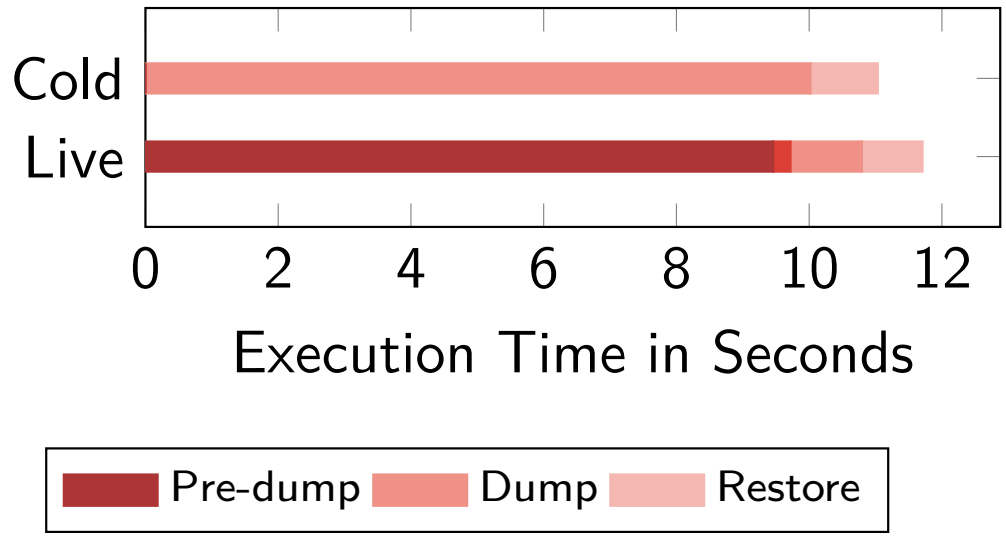
Comparison: Execution time

- libvirt & lxctools vs. LXC CLI
- Averaged over **200 runs**
- Save / restore with **1 GiB memory load**
- Libvirt support by the **cost** of a **few tens of milliseconds**




Empty VM / Container
Migration Time in s
0
1
2
3
4
0.25
0.5
1
2
4
Domain size in GiB
Migration with Workload
Migration Time in s
0
10
20
30
40
Min
0.25
0.5
1
2
4
Workload in GiB
lxctools
QEMU / KVM

- **1 GiB memory load**
- Averaged over **20 runs**
- In this case:
  only **two pre-dumps**
- **Live Migration**
  Downtime of **1.96 s**



- **First** libvirt driver **with migration support for LXC** containers
- **Working prototype**
  - Support for container management via libvirt
  - Minimal overhead to LXC CLI
  - Promising migration times
- **Future work**
  - Reduction of downtimes for live-migration
  - RDMA migration
- **Open source:** `https://github.com/RWTH-OS/libvirt`

Thank you for your kind attention!

Simon Pickartz et al. – spickartz@eonerc.rwth-aachen.de

Institute for Automation of Complex Power Systems
E.ON Energy Research Center, RWTH Aachen University
Mathieustraße 10
52074 Aachen

www.eonerc.rwth-aachen.de