Containers in a File

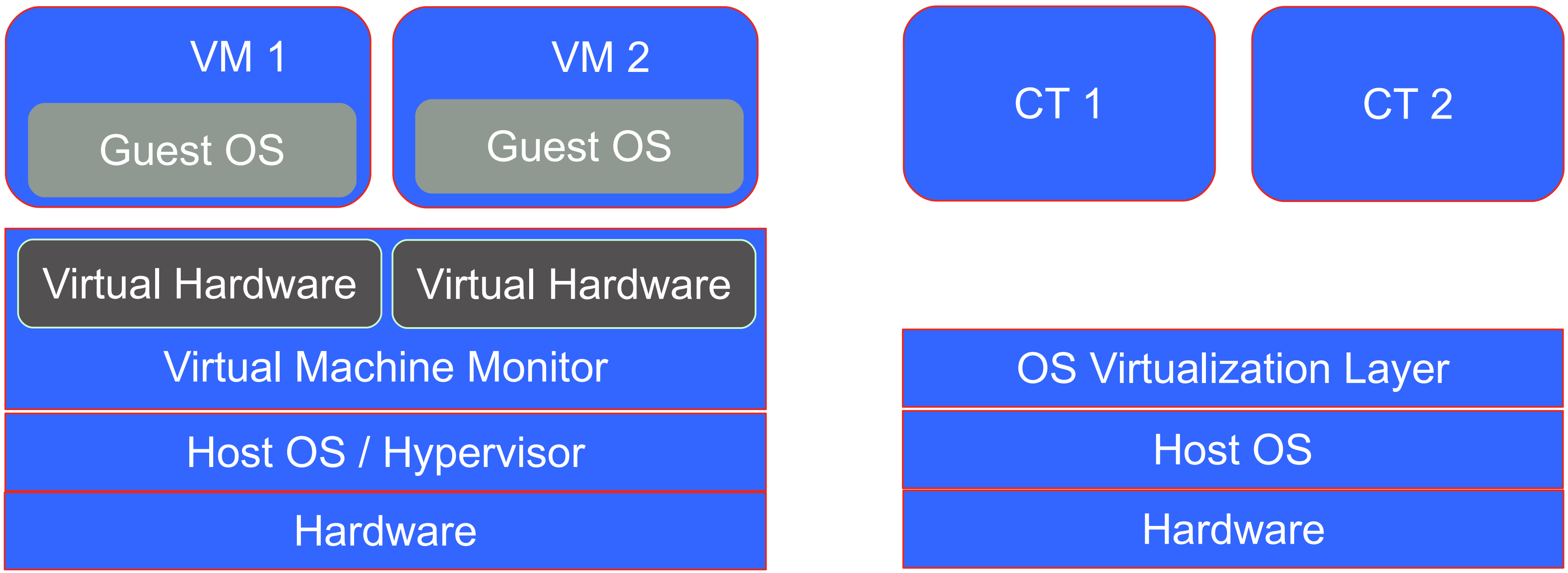
Profit from the Cloud

Maxim Patlasov
<mpatlasov@parallels.com>

Agenda

- Containers overview
- Current way of keeping container files and its problems
- Basics of container-in-a-file approach
- Limitations of existing facilities
- Solution
- Benefits
- Plans for future

Virtualization: VM-style vs. containers



VMMs:
- KVM / Qemu
- VMware
- Parallels Hypervisor

Containers:
- OpenVZ / Parallels Containers
- Solaris containers/Zones
- FreeBSD jails

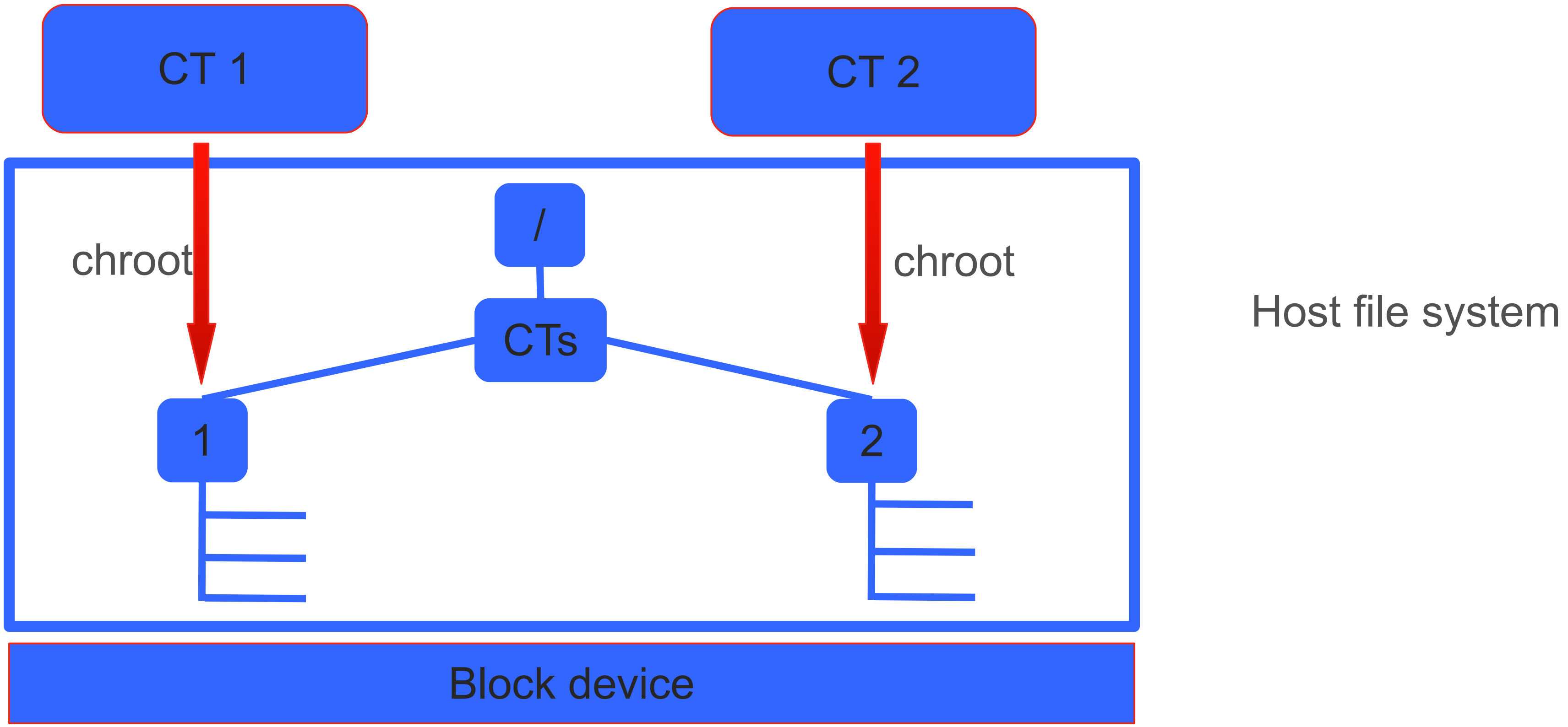
Containers Virtualization

Each container has its own:

- **Files**
- **Process tree**
- **Network**
- **Devices**
- **IPC objects**

Containers: file system view

Natural approach: each container chroot-s to a per-container sub-tree of some system-wide host file system.

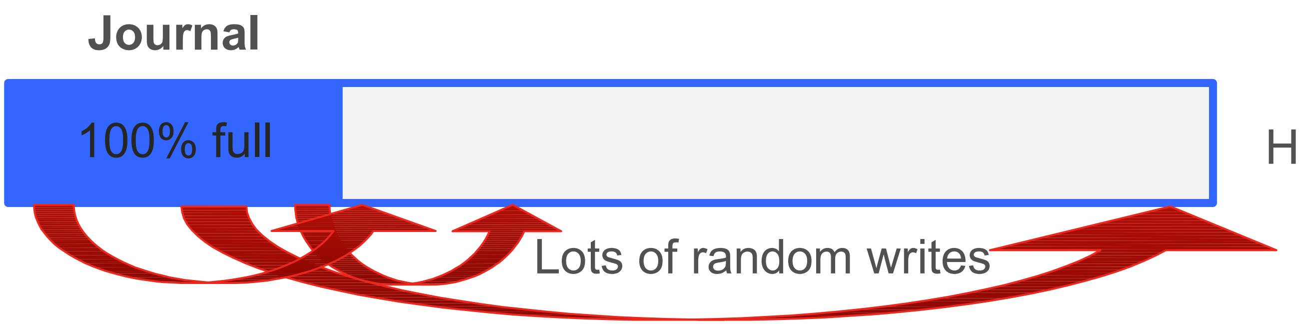


Profit from the Cloud

Problems

· File system journal is a bottleneck:

1. CT1 performs a lot of operations leading to metadata updates (e.g. truncates). Consequently, the journal is getting full.



2. CT2 issuing a single I/O operation blocks until journal checkpoint is completed. Up to **15** seconds in some tests!

Problems (cont.)

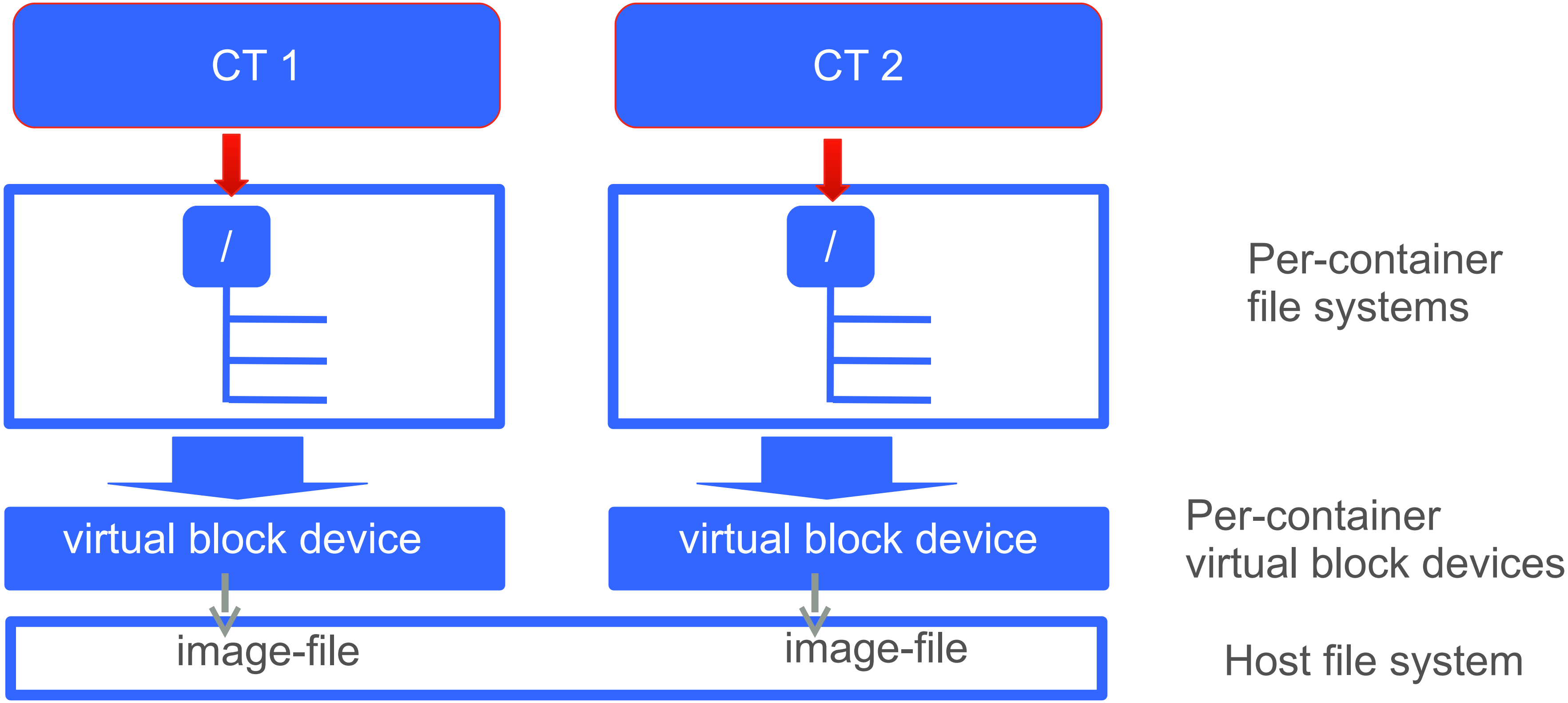
- Lots of small-size files I/O on any CT operation (backup, clone)
- Sub-tree disk quota support is absent in mainstream kernel

- Hard to manage:
  - No per-container snapshots, live backup is problematic
  - Live migration – rsync unreliable, changed inode numbers

- File system type and its properties are fixed for all containers
- Need to limit number of inodes per container

Container in a file

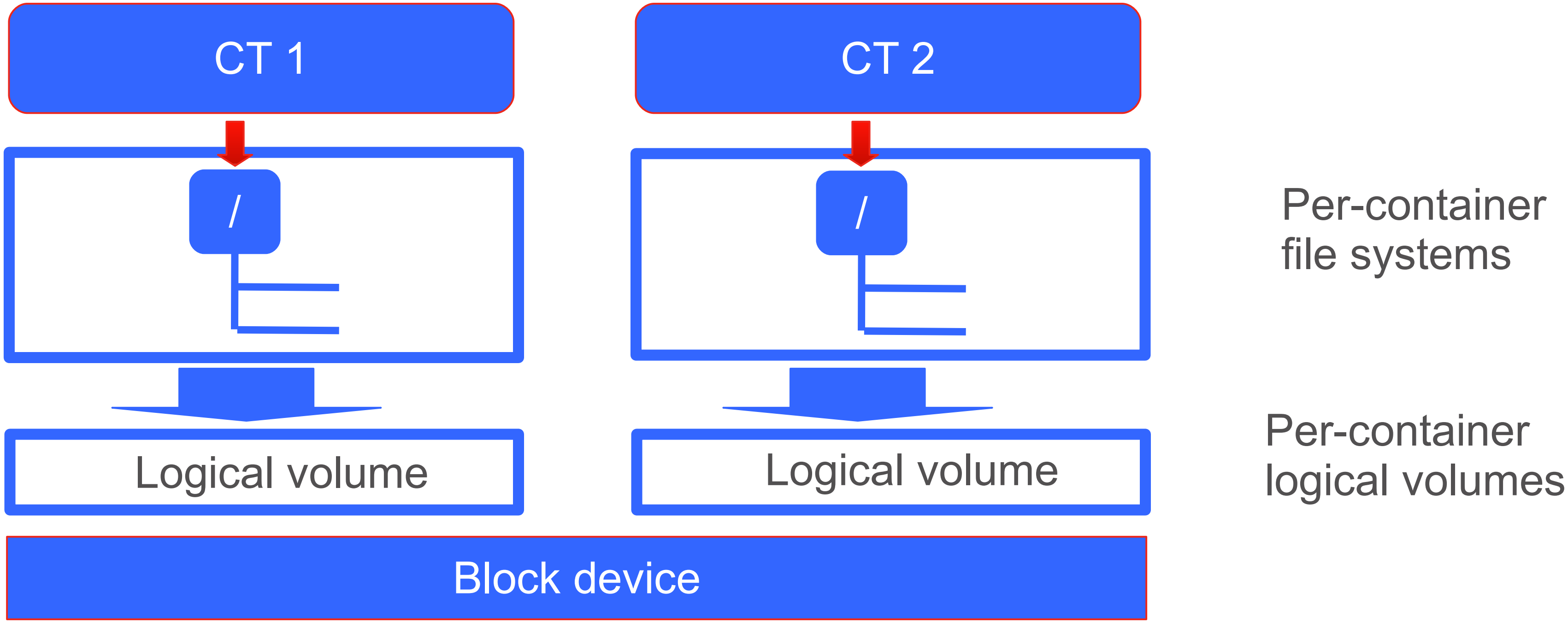
Basic idea: assign virtual block device to container, keep container' file-system on top of virtual block device.



Profit from the Cloud

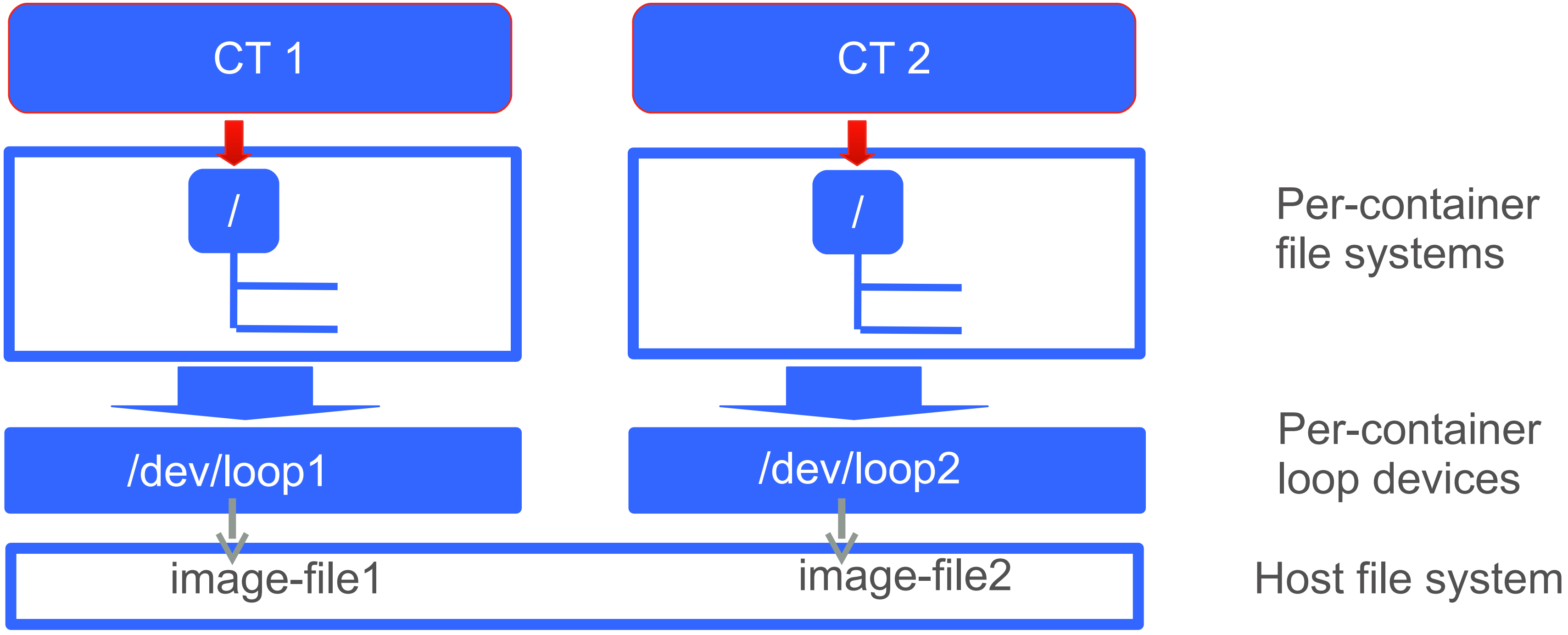
LVM limitations

- Not flexible enough – works only on top of block device
- Hard to manage (e.g. how to migrate huge volume?)
- No dynamic allocation (--virtualsize 16T --size 16M)
- Management is not as convenient as for files

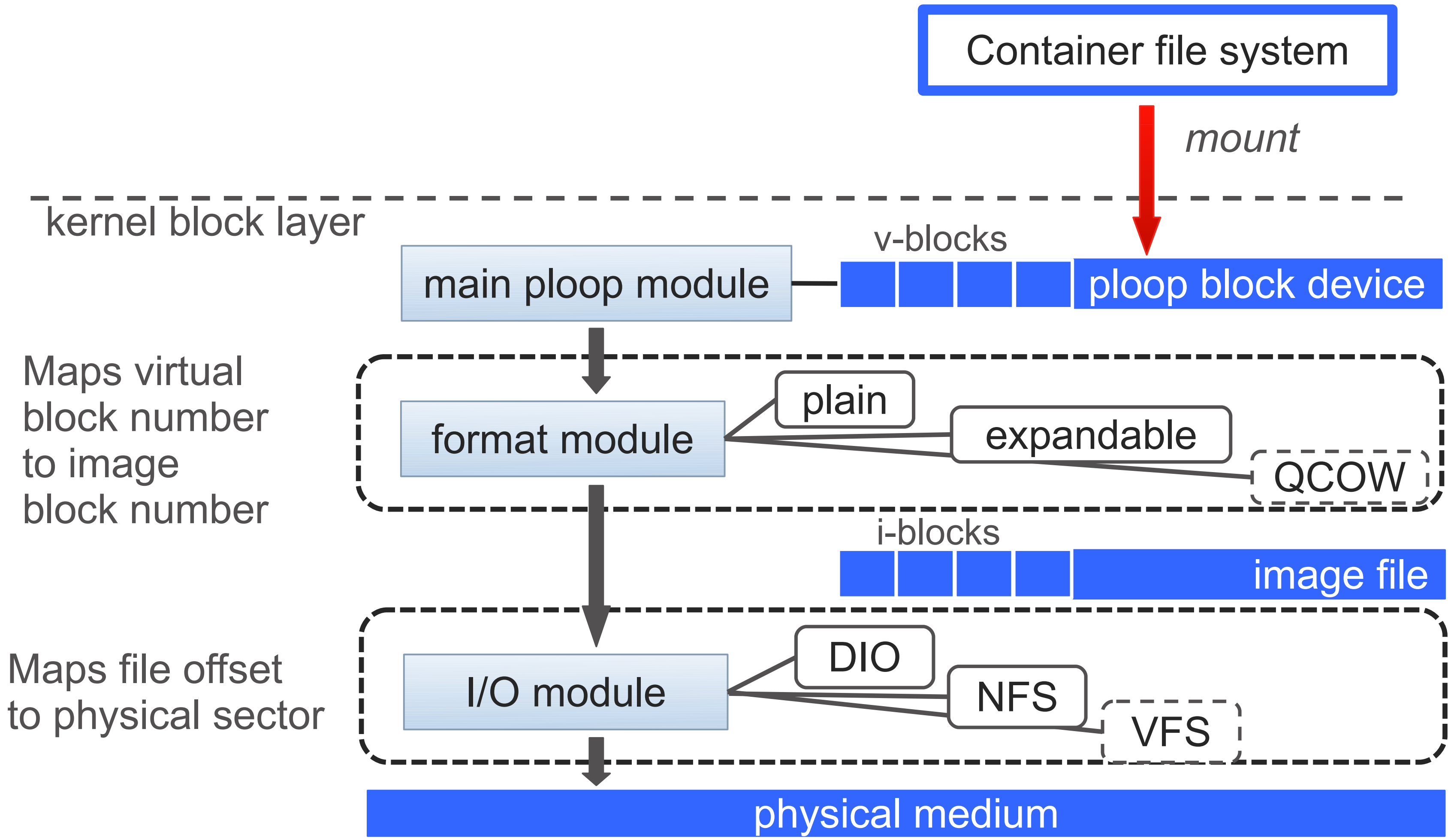


Ordinary loop device limitations

- VFS operations leads to double page-caching
- No dynamic allocation (only “plain” format is supported)
- No helps to backup/migrate containers
- No snapshot functionality



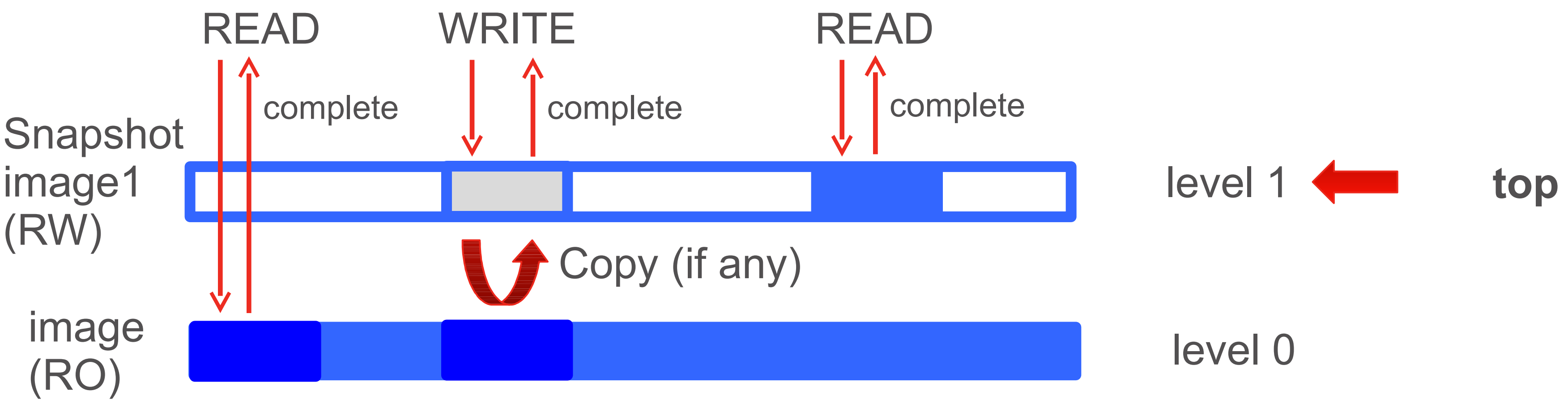
Design of new loop device (ploop)



Stacked image configuration

Ploop snapshots are represented by image files. They are stuffed in ploop device in stacked manner and obey the following rules:

- Only top image is opened in RW mode (others – RO)
- Every mapping bears info about ‘level’
- READ leads to access to an image of proper ‘level’
- WRITE may lead to transferring proper block to top image



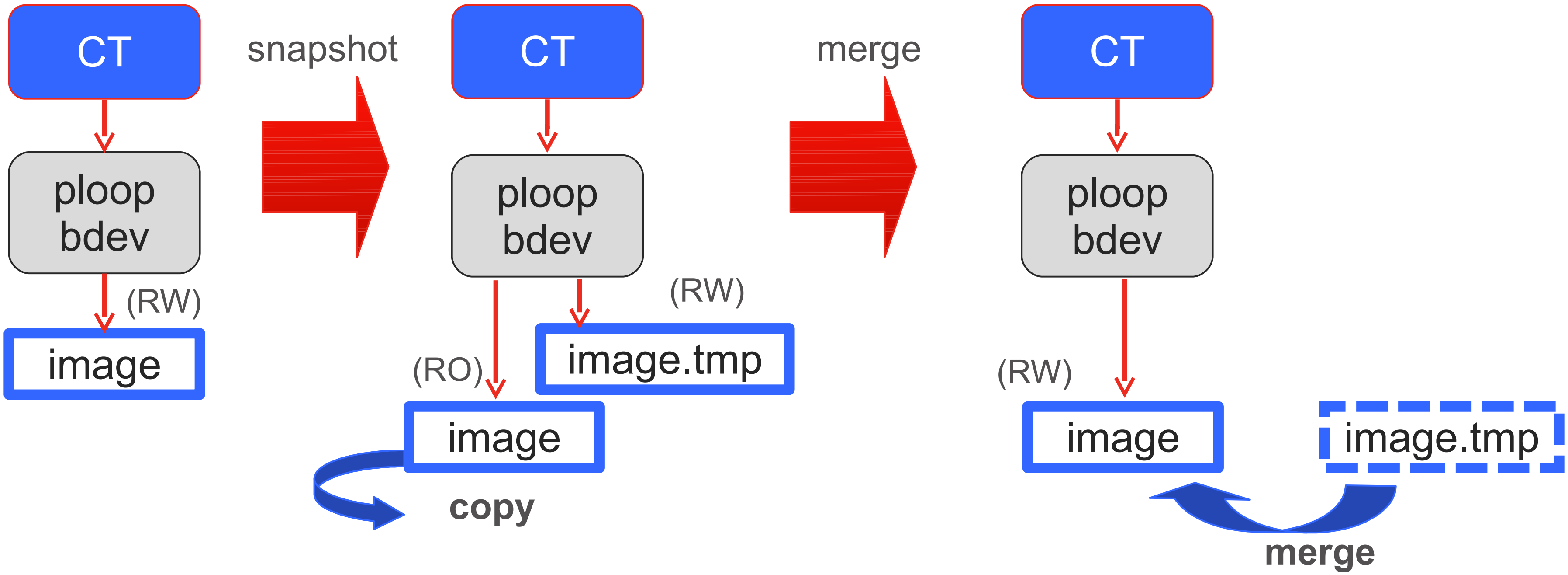
Backup via snapshot

Operations involved:

- Create snapshot
- Backup base image
- Merge

Key features:

- **On-line**
- **consistent**



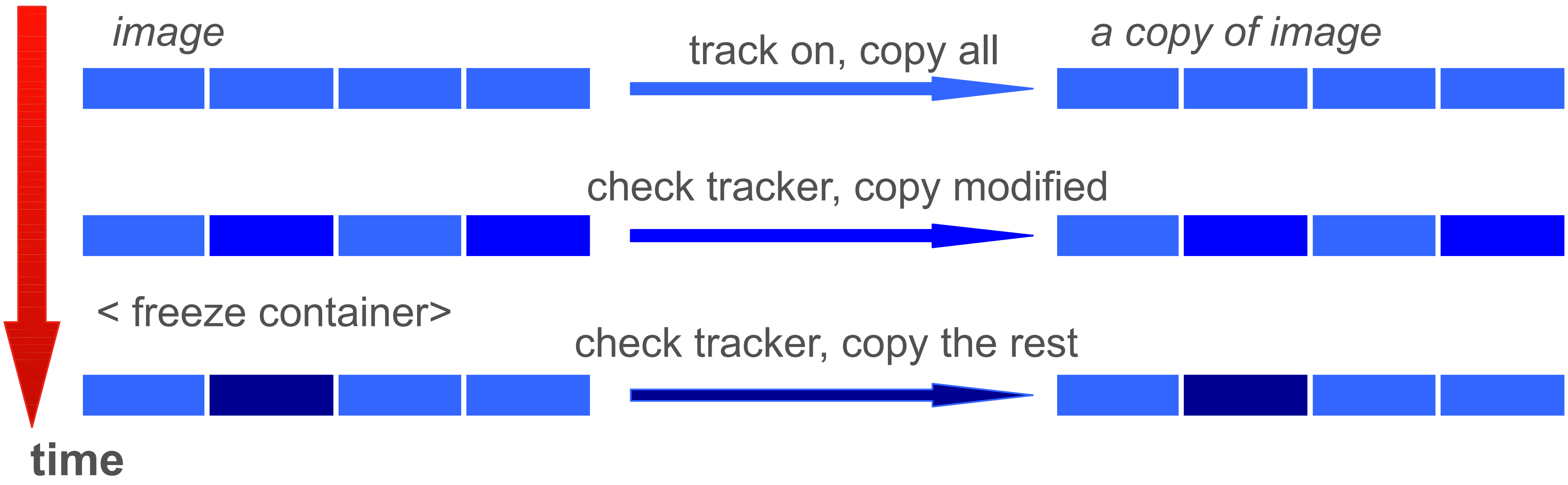
Migrate via write-tracker

Operations involved:

- Turn write-tracker on
- Iterative copy
- Freeze container
- Copy the rest

Key features:

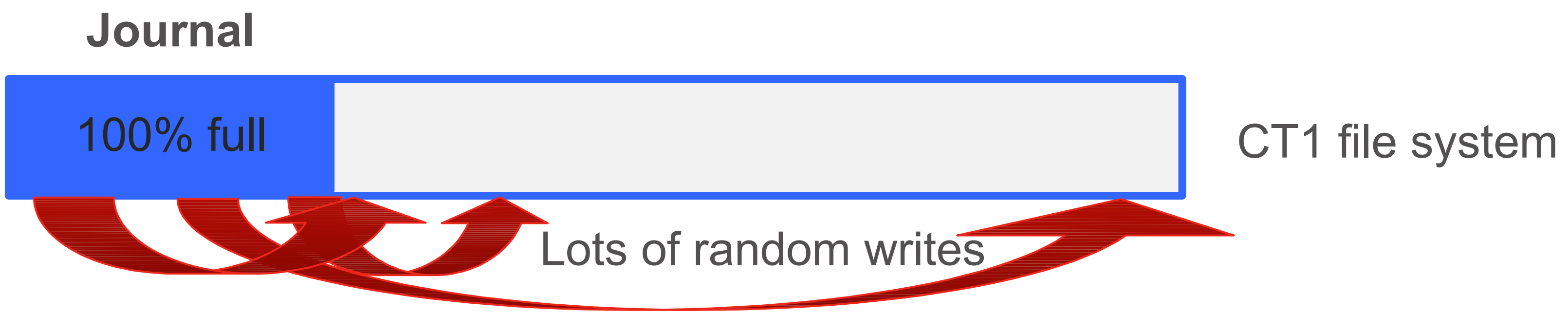
- **On-line**
- **I/O efficient**



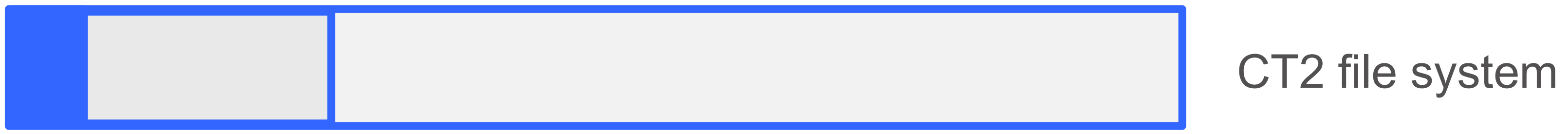
Problems solved

- File system journal is not bottleneck anymore

1. CT1 performs a lot of operations leading to metadata updates (e.g. truncates). Consequently, the journal is getting full.



2. CT2 has its own file system. So, it's not directly affected by CT1.



Problems solved (cont.)

- Large-size image files I/O instead of lots of small-size files I/O on management operations
- Disk quota can be implemented based on virtual device sizes. No need for sub-tree quotas
- Live backup is easy and consistent
- Live migration is reliable and efficient
- Different containers may use file systems of different types and properties
- No need to limit “number-of-inodes-per-container”

Additional benefits

- Efficient container creation
- Snapshot/merge feature
- Can support QCOW2 and other image formats
- Can support arbitrary storage types
- Can help KVM to keep all disk I/O in-kernel

Plans

- Implement I/O module on top of vfs ops
- Add support of QCOW2 image format
- Optimizations:
- discard/TRIM events support
- online images defragmentation
- support sparse files
- Integration into mainline kernel

Questions