



QEMU Backup

Maxim Nestratov, Virtuozzo

Vladimir Sementsov-Ogievskiy, Virtuozzo

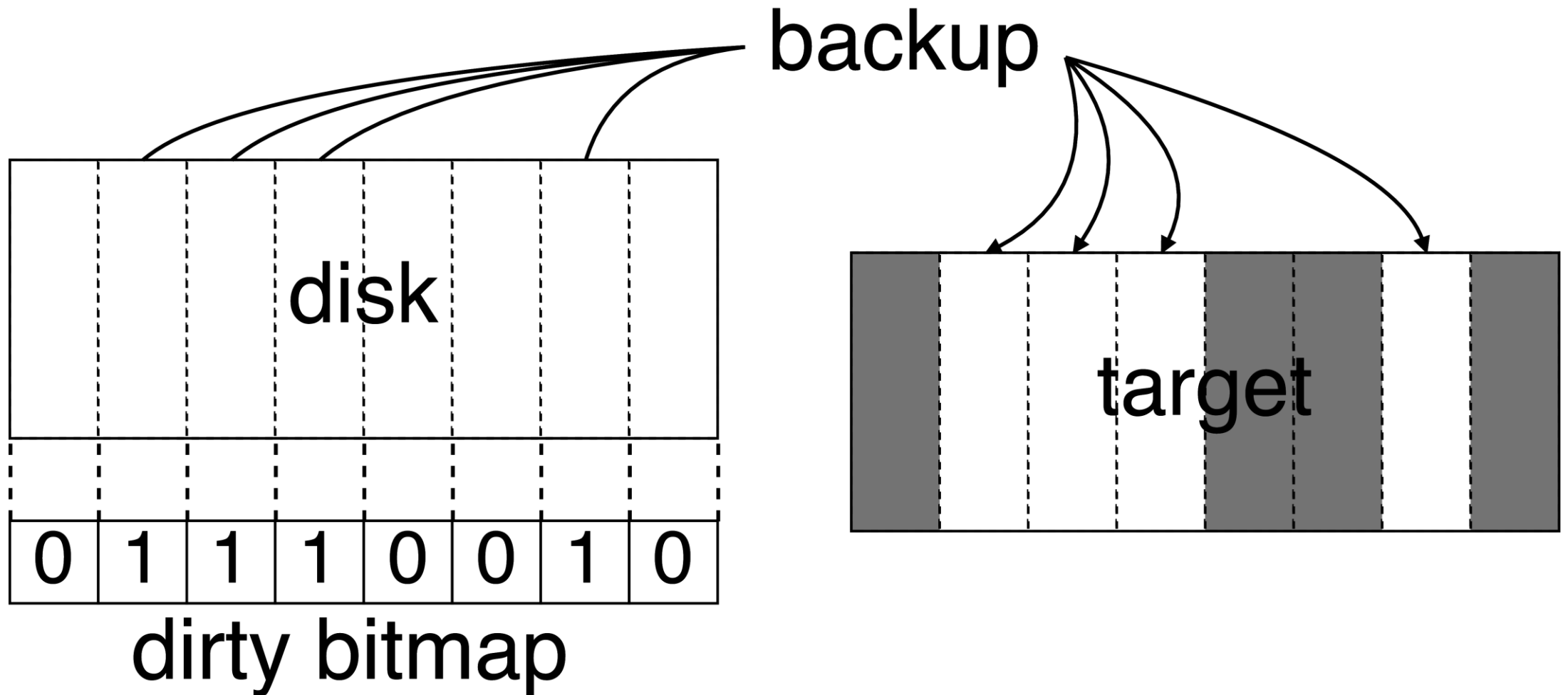
QEMU Backup

Vladimir Sementsov-Ogievskiy, Virtuozzo

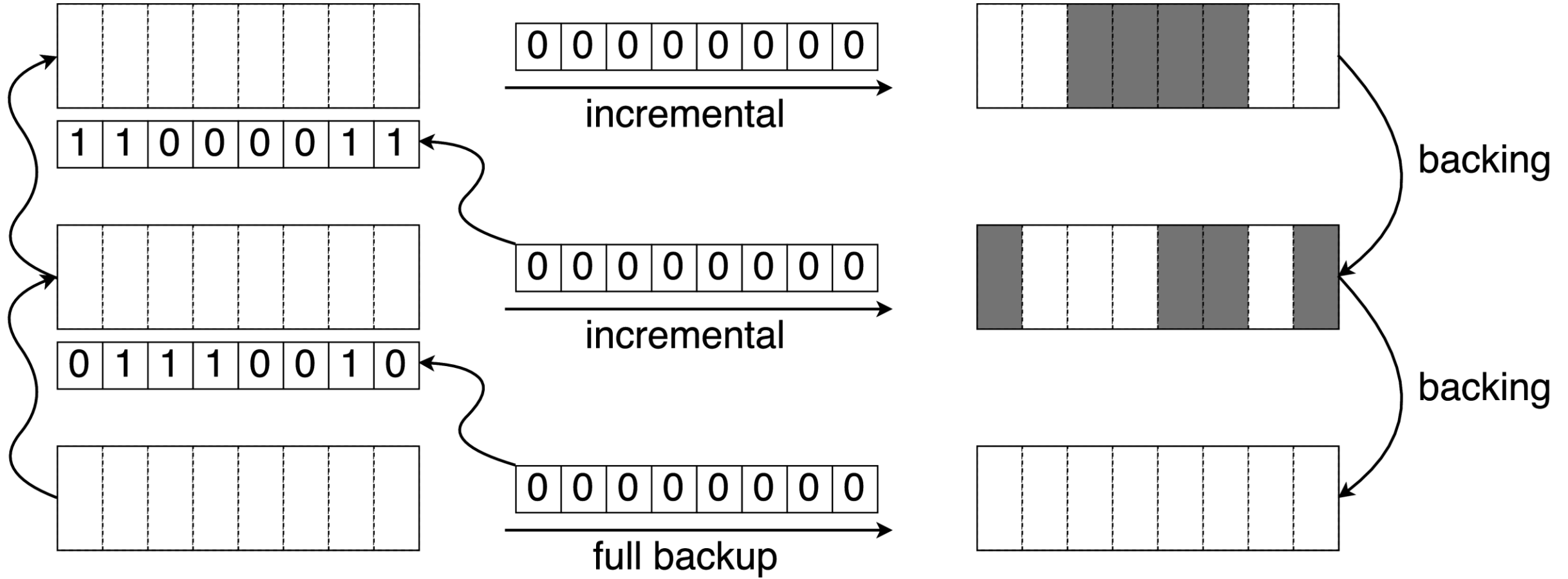
Full featured backup

- Online backup
 - Fast
 - Not very invasive for the guest
- Incremental
 - Dirty bitmaps persistence and migration
- External backup API

Incremental



Incremental



Incremental: persistent

- Qcow2 bitmaps merged into 2.9
 - Several named bitmaps
 - Size equals image size
 - Sparse format
- Other formats are under discussion

Incremental: migration

Variants:

- First approach: meta bitmaps
- Current approach: postcopy
- Through storage (works for qcow2)

Incremental: snapshots

Internal snapshots

- Dirty bitmaps correspond to active state
- Switch to snapshot = the whole disk is dirty

External snapshots

- Everything is possible

Performance: current work

Backup = simple copy + COW (write notifiers)

- Current approach:
 - Sequential copying + sequential notifiers
- New arc:
 - Queues of requests
 - Several copying coroutines
 - Notifiers just increase priority of request
 - Earlier notifier release

Performance: ideas and plans

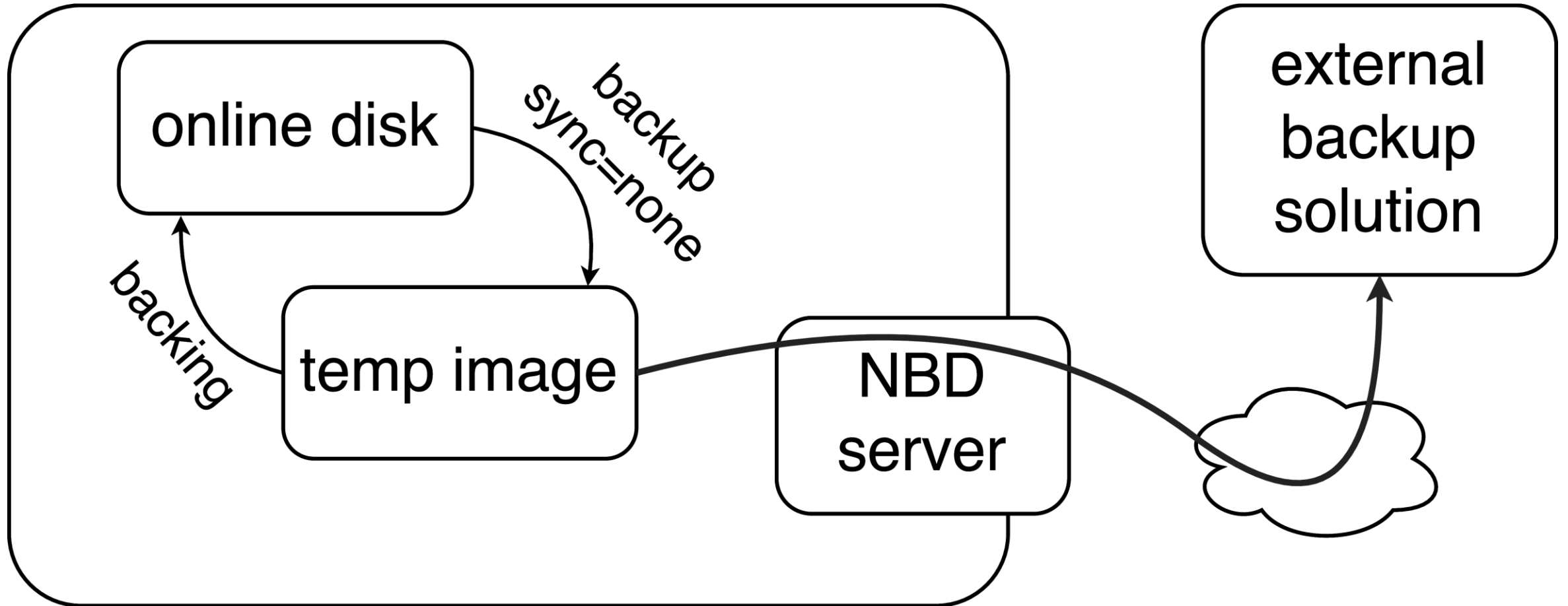
How to handle COW?

- Current: guest wait for backup
- Reverse delta: read COW area to local delta

External backup API

- Image fleecing scheme
- Incremental backup
 - NBD block-status extension
 - Additional API for dirty bitmap management

External backup API: image fleecing



External backup API: NBD block-status

- Current NBD: payload only for READ
- Extension: structured replies
- Extension: block-status
 - Negotiation phase: select metadata contexts
 - Transmission phase
 - New command NBD_CMD_BLOCK_STATUS
 - Reply chunk contains extent descriptors

QEMU Backup summary

Merged:

- Qcow2 bitmaps

Done in Virtuozzo:

- New backup architecture (async IO)
- Bitmaps migration
- NBD block-status extension

Near future:

- External backup API

Libvirt Backup

Maxim Nestratov, Virtuozzo

Libvirt Backup API first proposal

- "Push" backups
- "Pull" backups

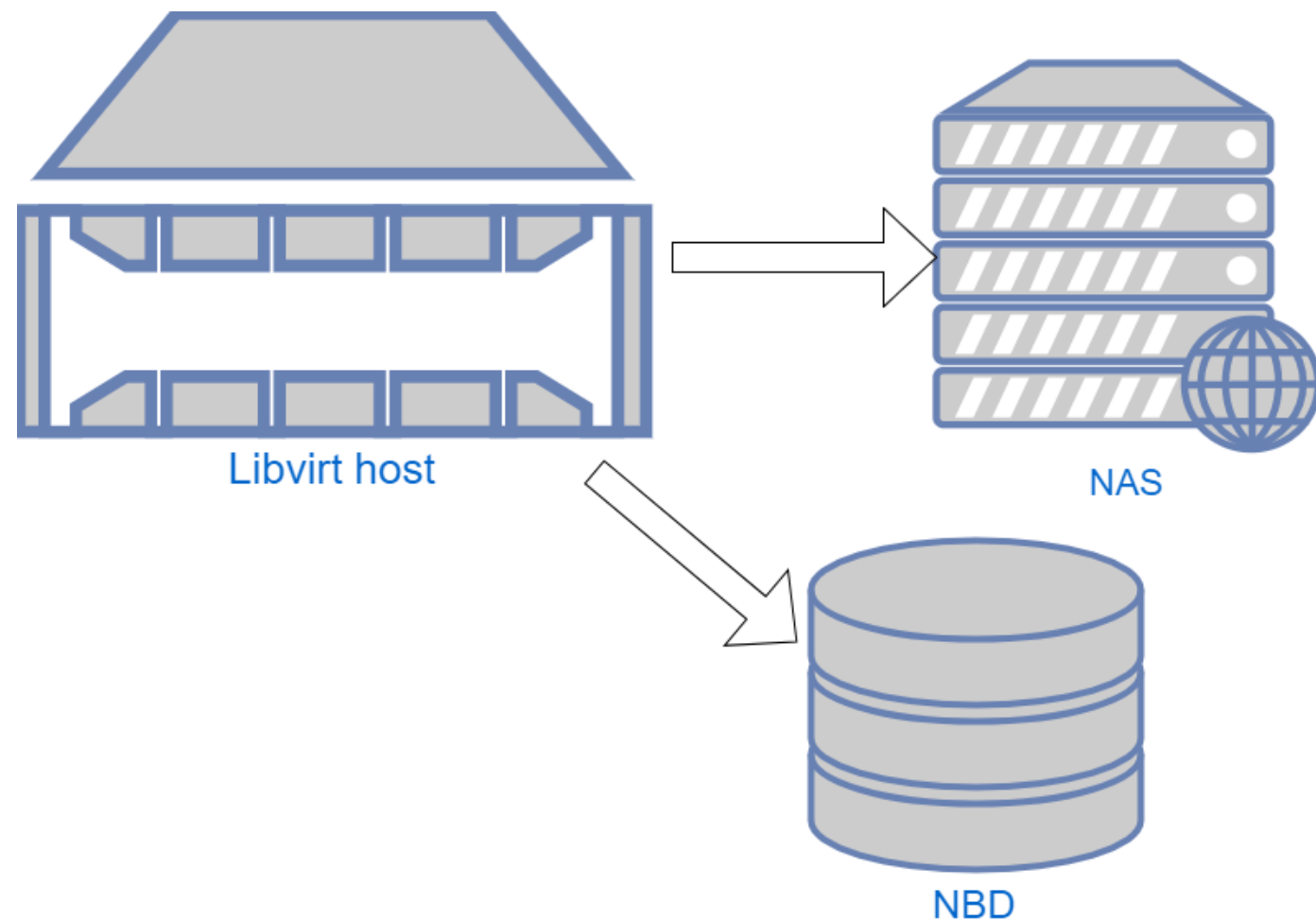
Why not snapshots?

- Different storage
- Incremental backups
- Multiple chains
- Agentless

"Push" or "Managed" Backups

- A new set of functions similar to snapshots
- Create, List, Delete, Edit
- Managed by libvirt
- Local to host
- Can be saved to any supported block device
- NAT friendly

Managed backup scheme



Managed backup concerns

- Hard to use in clusters
- Guest performance influence due to network throughput
- A lot of code to implement
- Access to backup storage

"Pull" or "External" Backups

- Mostly two functions: Start/Stop
- Exposes block device via NBD protocol
- Uses NBD protocol extension for incremental backups

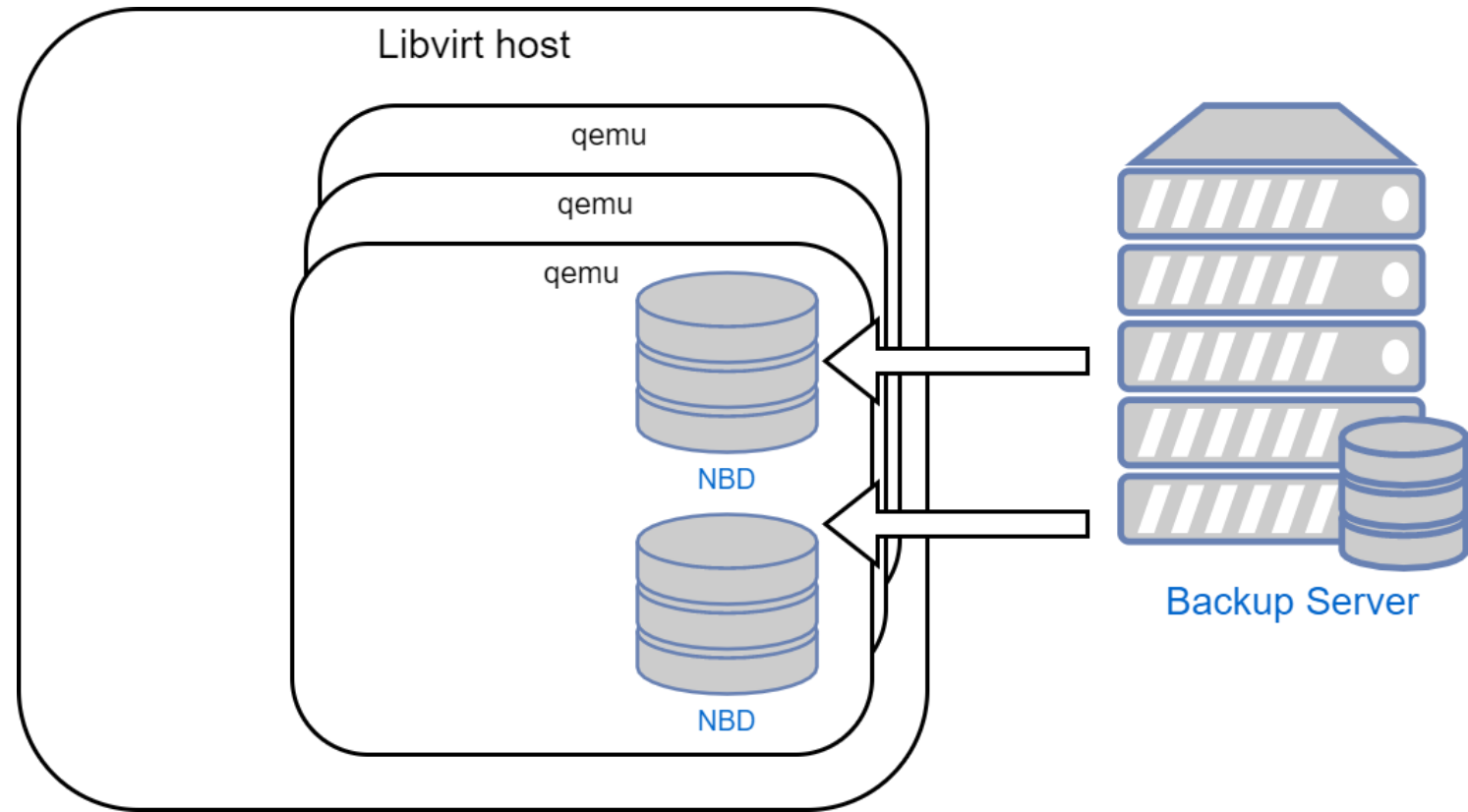
External backups advantages

- Tolerant to guests performance
- Controlled externally
- Cluster friendly

External backups concerns

- NAT unfriendly
- Tolerant to guests performance
- Uses NBD protocol extension for incremental backups
- Controlled externally
- Not bound to a specific hypervisor host

External backup scheme





Thank you

Maxim mnestratov@virtuozzo.com

Vladimir vsementsov@virtuozzo.com

