Optimizing VM Checkpointing for Restore Performance in VMware ESXi Server

Irene Zhang
University of Washington

Tyler Denniston
MIT CSAIL

Yury Baskakov
VMware

Alex Garthwaite
CloudPhysics

Virtual Machine Checkpointing

- Provides snapshot of a running virtual machine.
- Taking a checkpoint is fast,
  but restoring a checkpoint is slow.
- Traditionally used to support fault tolerance applications.

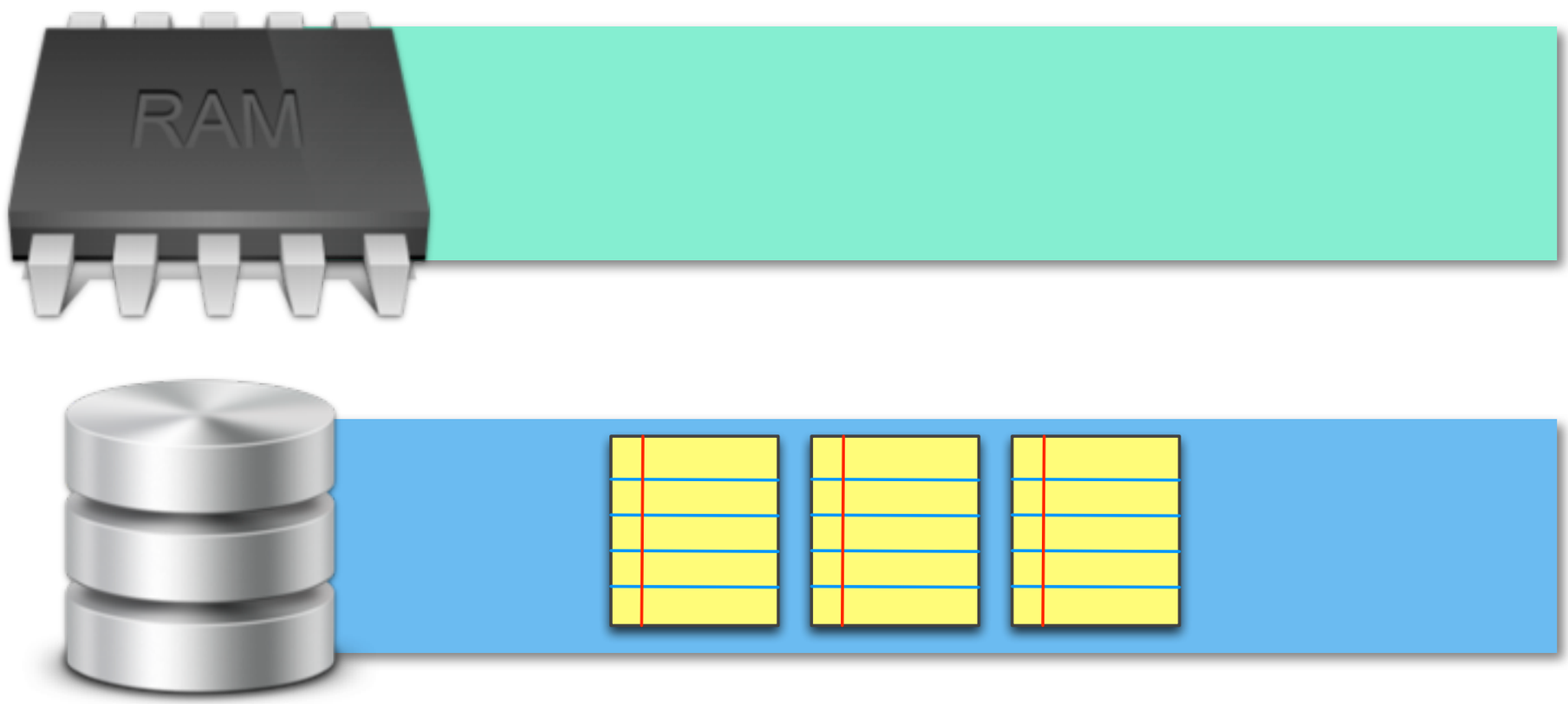
Existing VM checkpointing systems cannot support applications that require fast restore.

- Dynamic VM allocation
- Energy conservation
- Virtual desktop infrastructure

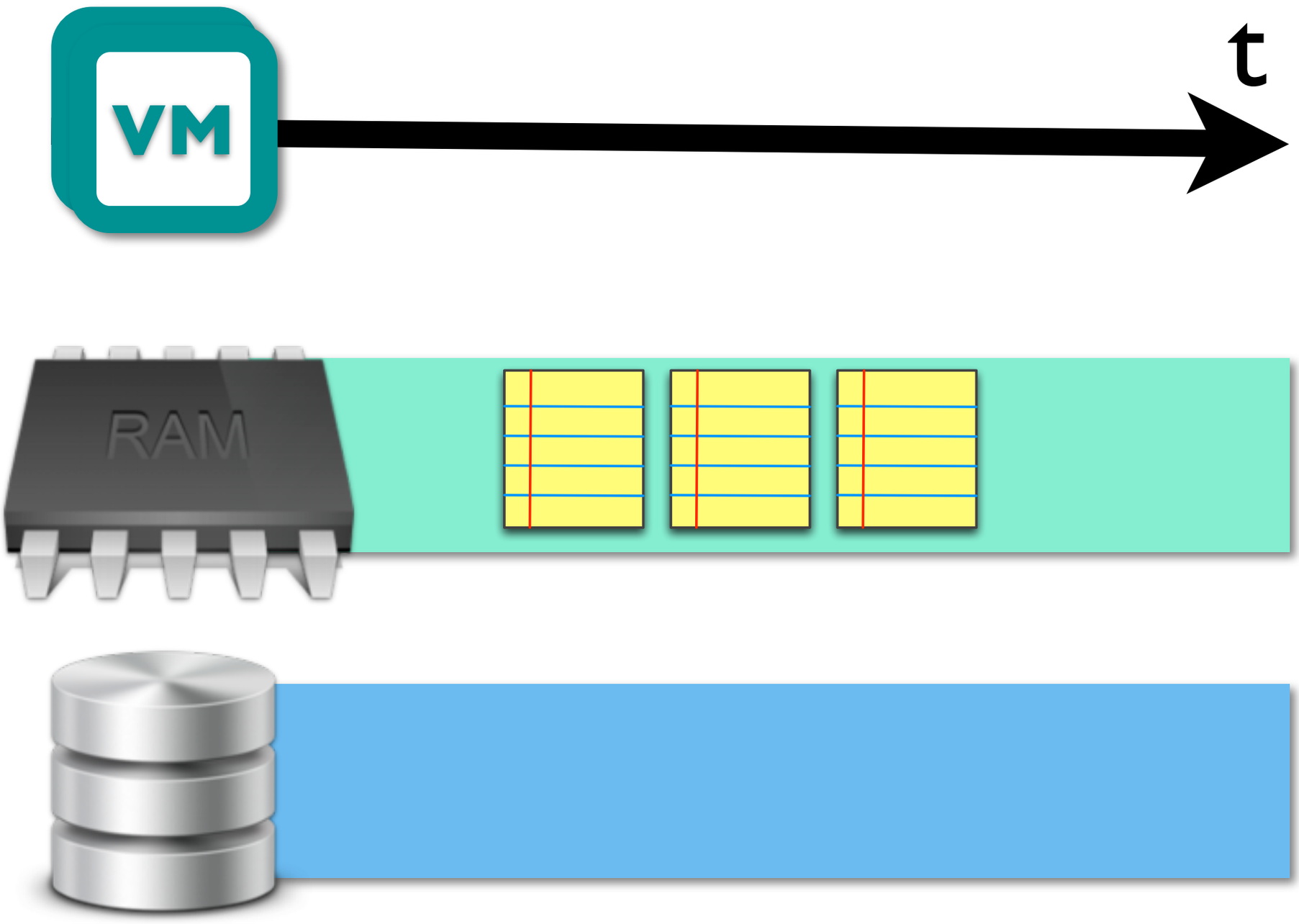
Halite

New checkpointing system for VMware ESXi Server that reduces restore time to seconds.

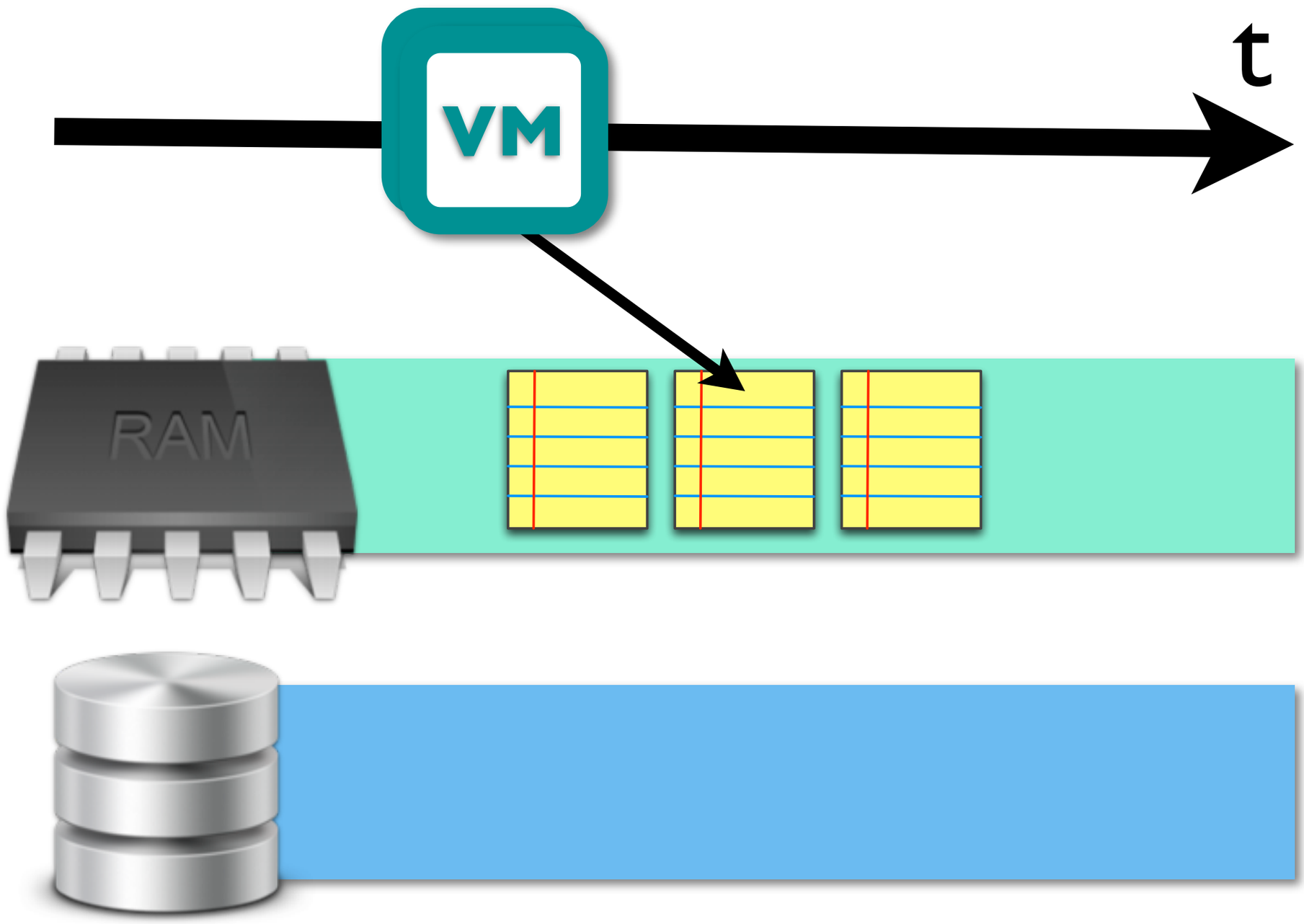
VM Checkpoint Restore



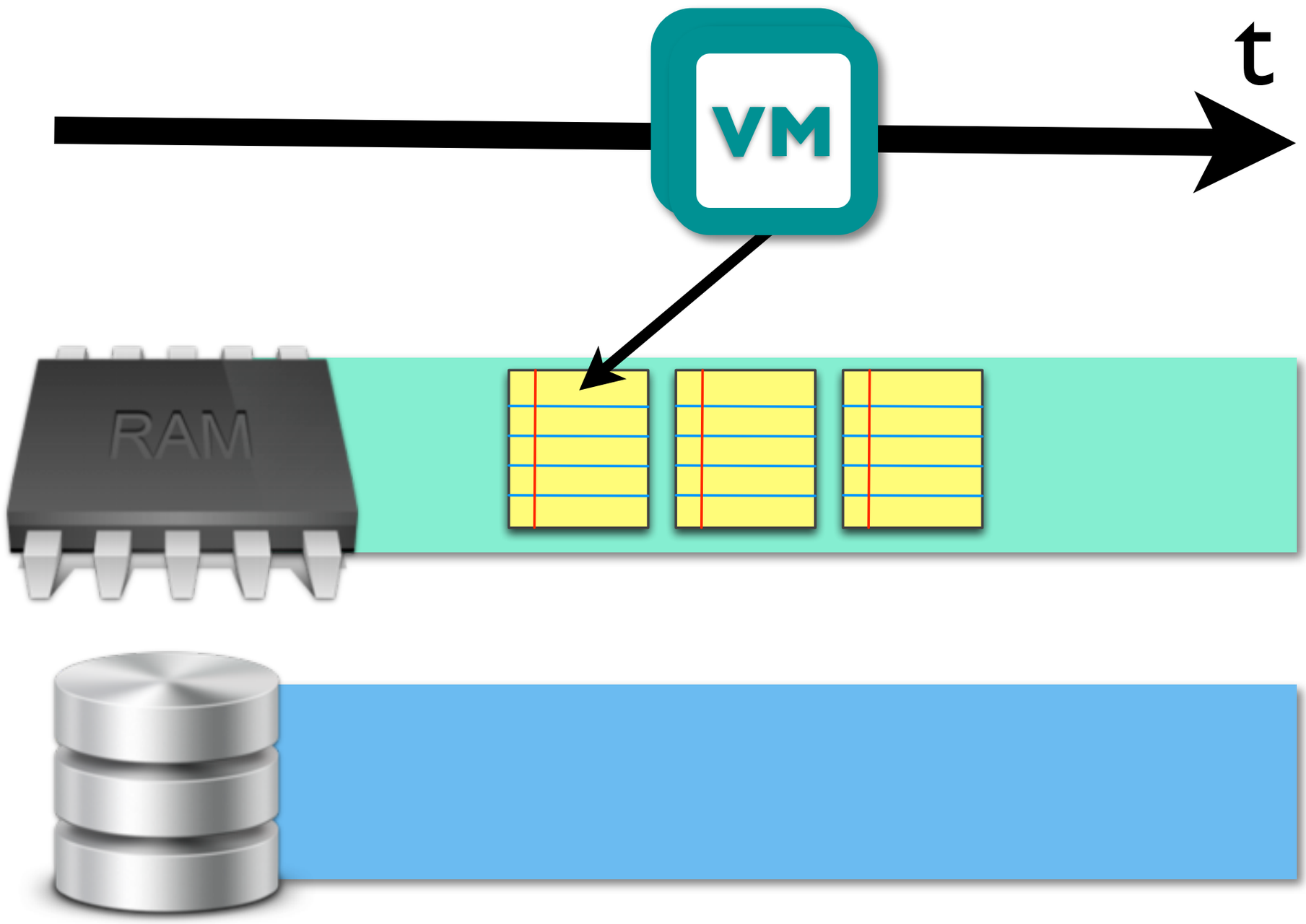
VM Checkpoint Restore



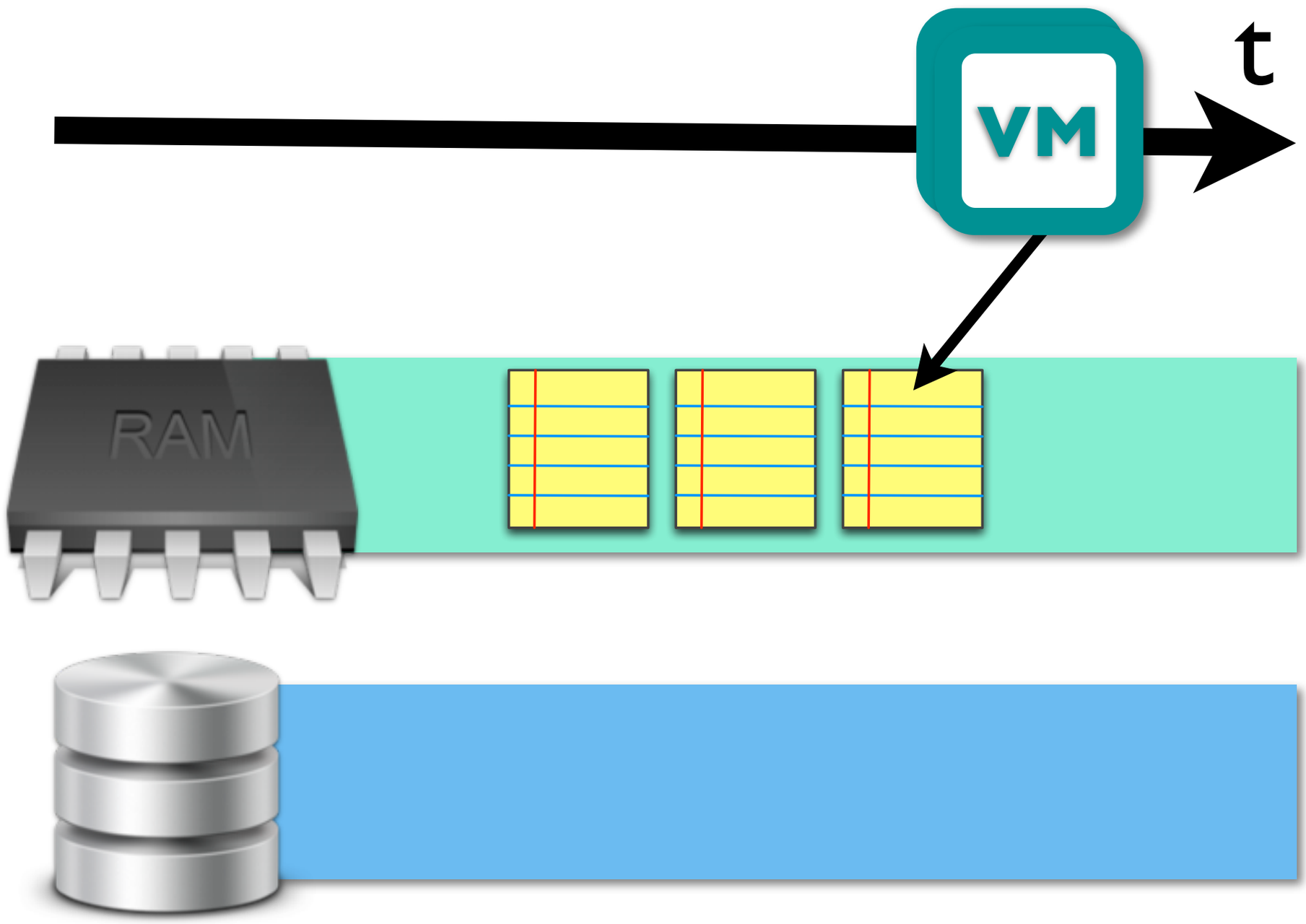
VM Checkpoint Restore



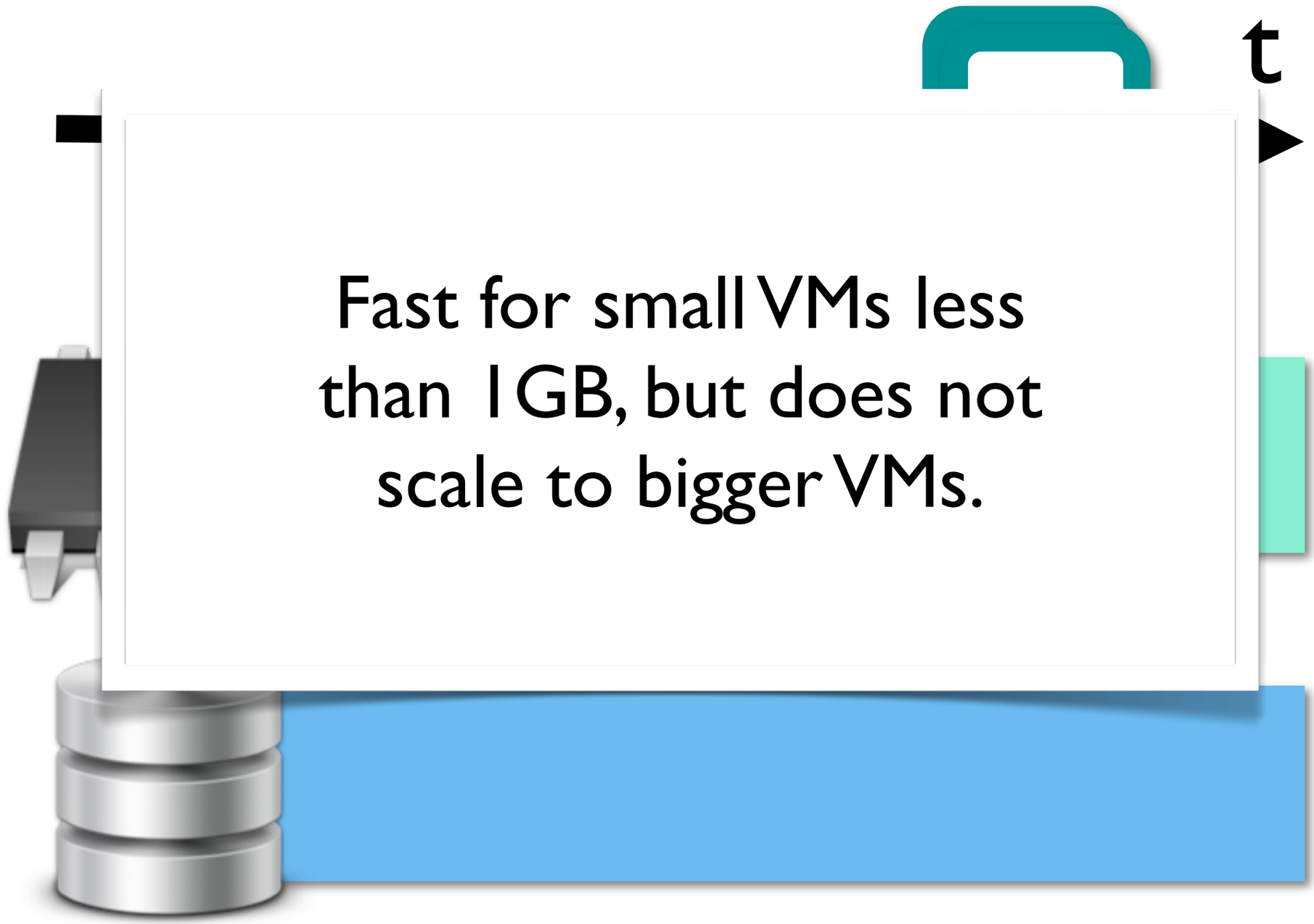
VM Checkpoint Restore



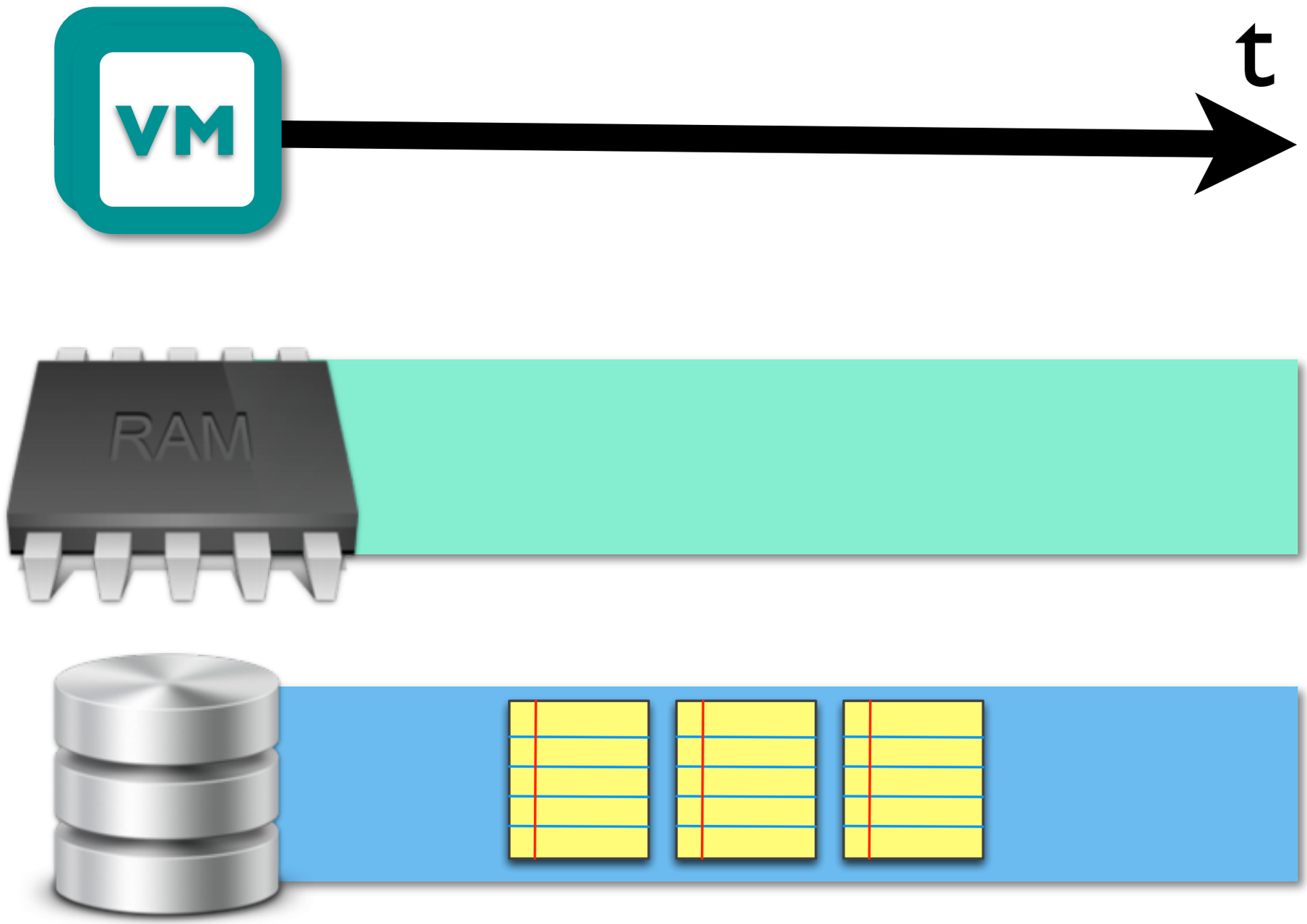
VM Checkpoint Restore



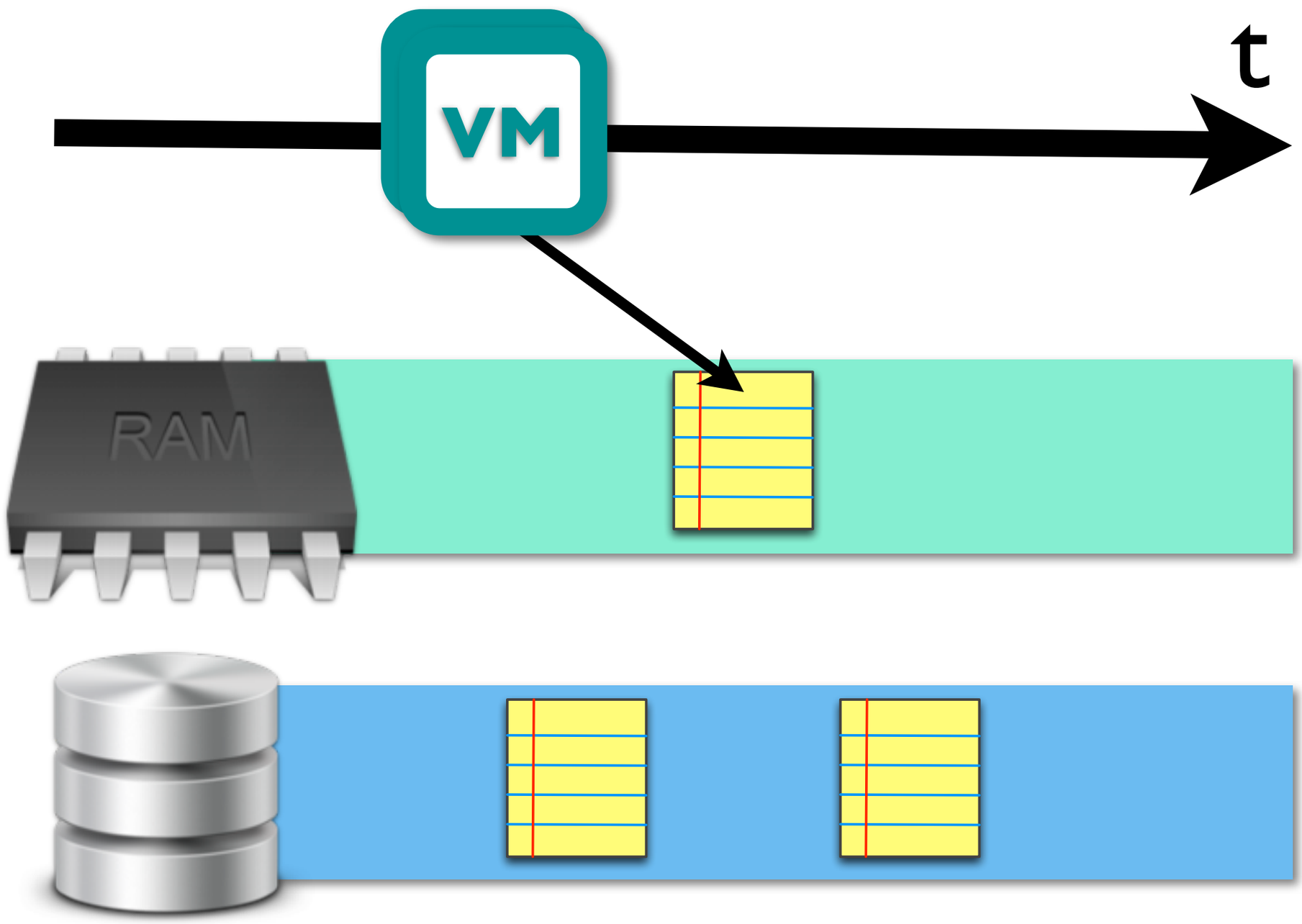
VM Checkpoint Restore



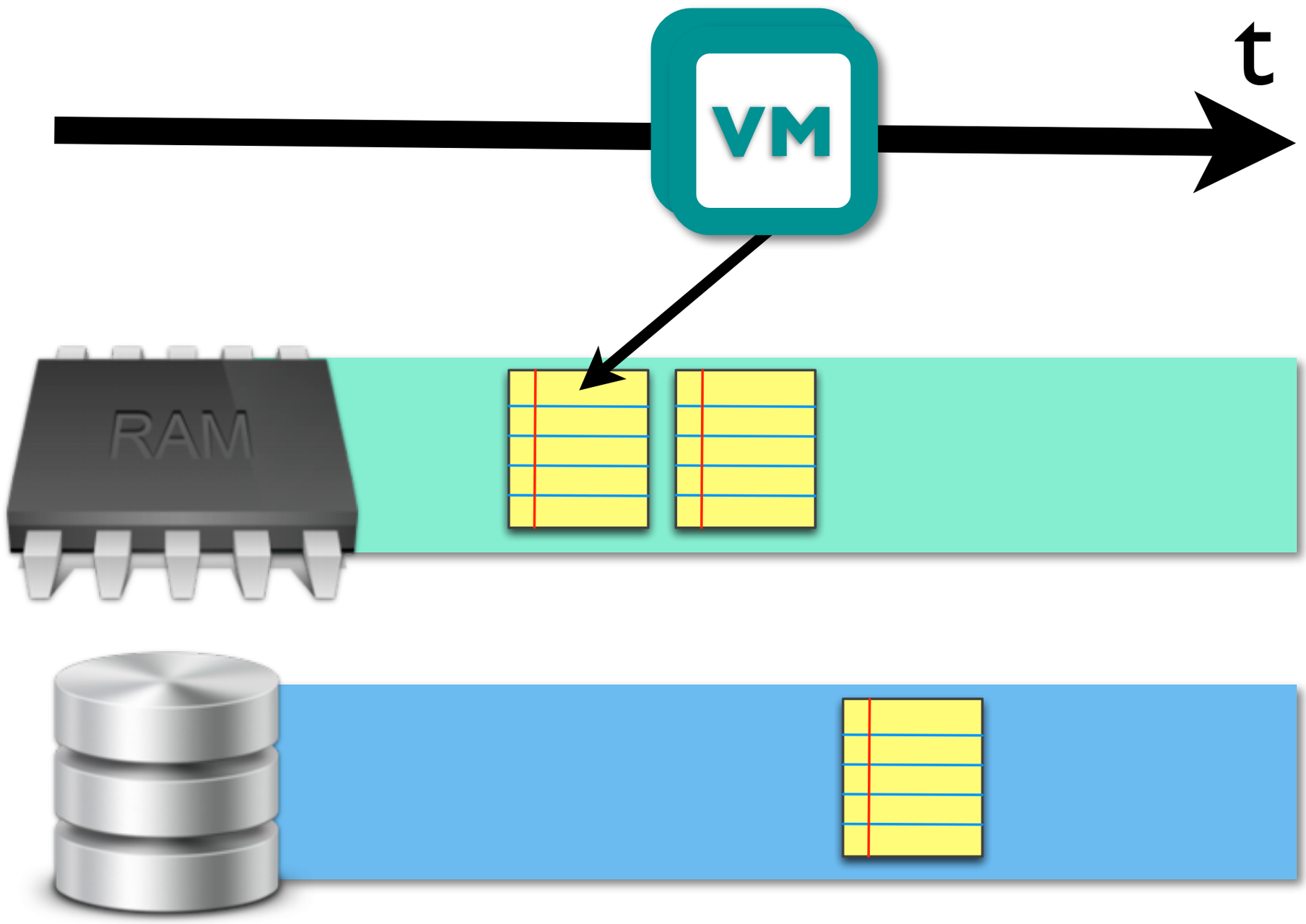
ESXi Checkpoint Restore



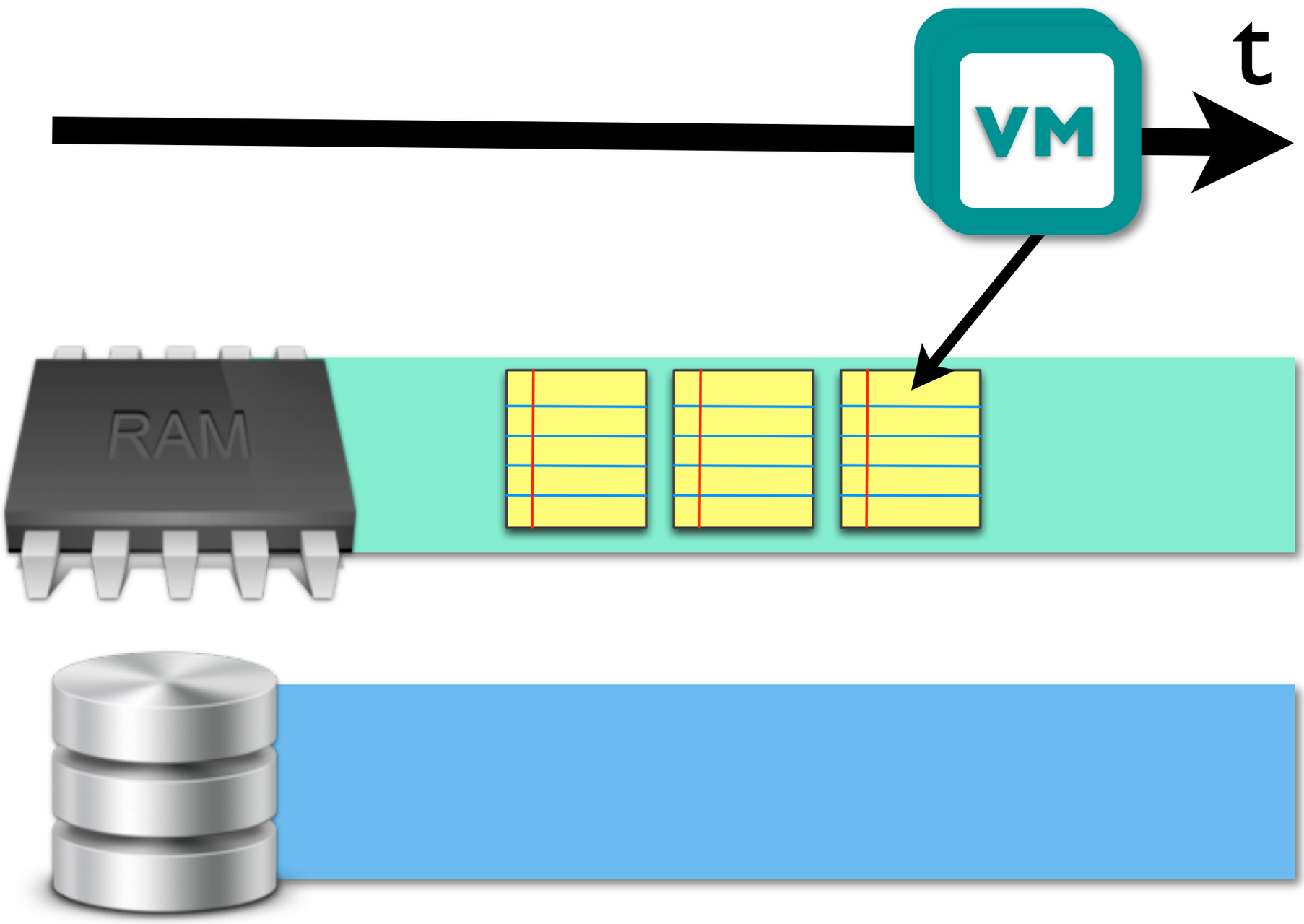
ESXi Checkpoint Restore



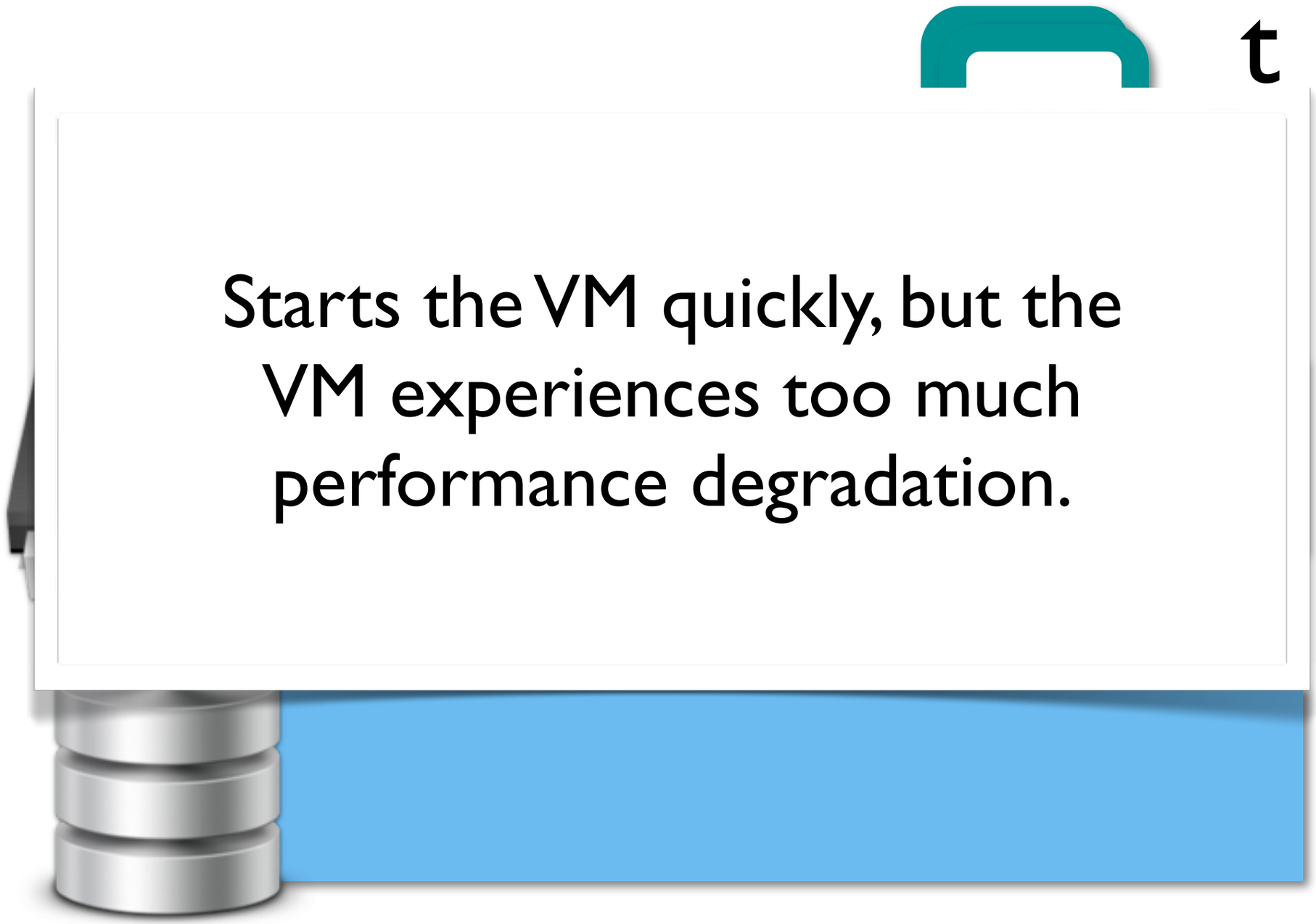
ESXi Checkpoint Restore



ESXi Checkpoint Restore



ESXi Checkpoint Restore



Faults to disk during checkpoint restore are inefficient.

- Checkpointed memory is organized in physical address order.
- Each fault is random 4KB disk read.
- No spatial locality, so bigger reads do not help.

Working Set Restore

[Zhang, et al., VEE 2011]

- Reduces faults to disk by prefetching pages the VM may access.
- Found some VM workloads are too complicated to predict.
- Many sources of divergence: timing variations, background processes and external/user inputs.

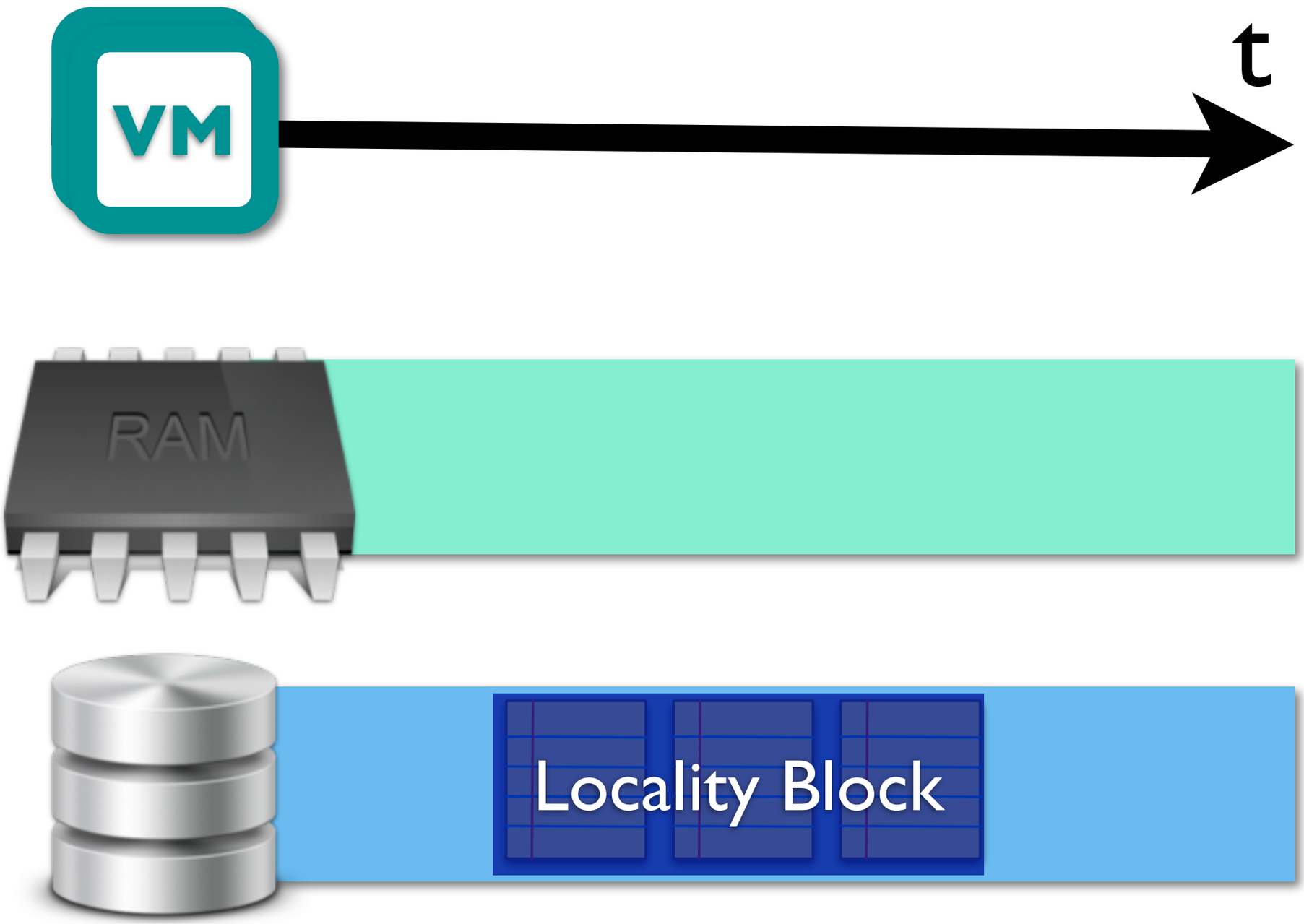
Halite predicts access locality.

- Predict pages that are likely to be accessed together and store them together.
- Sufficient to provide spatial locality in the checkpointing file, enabling bigger reads.
- Easier to predict and more resilient to divergence.

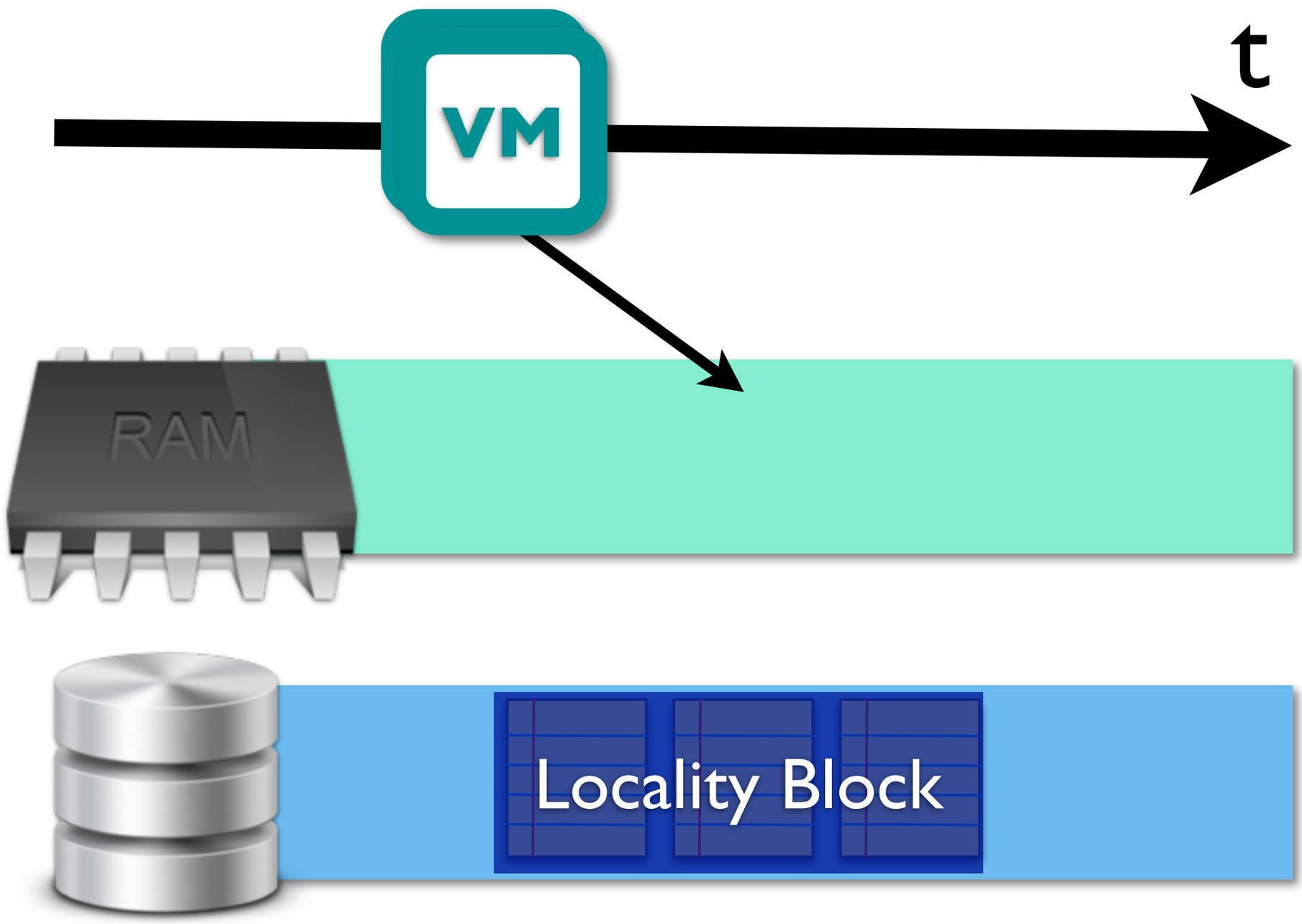
Locality Blocks

Block of checkpointed VM memory pages likely to be accessed together.

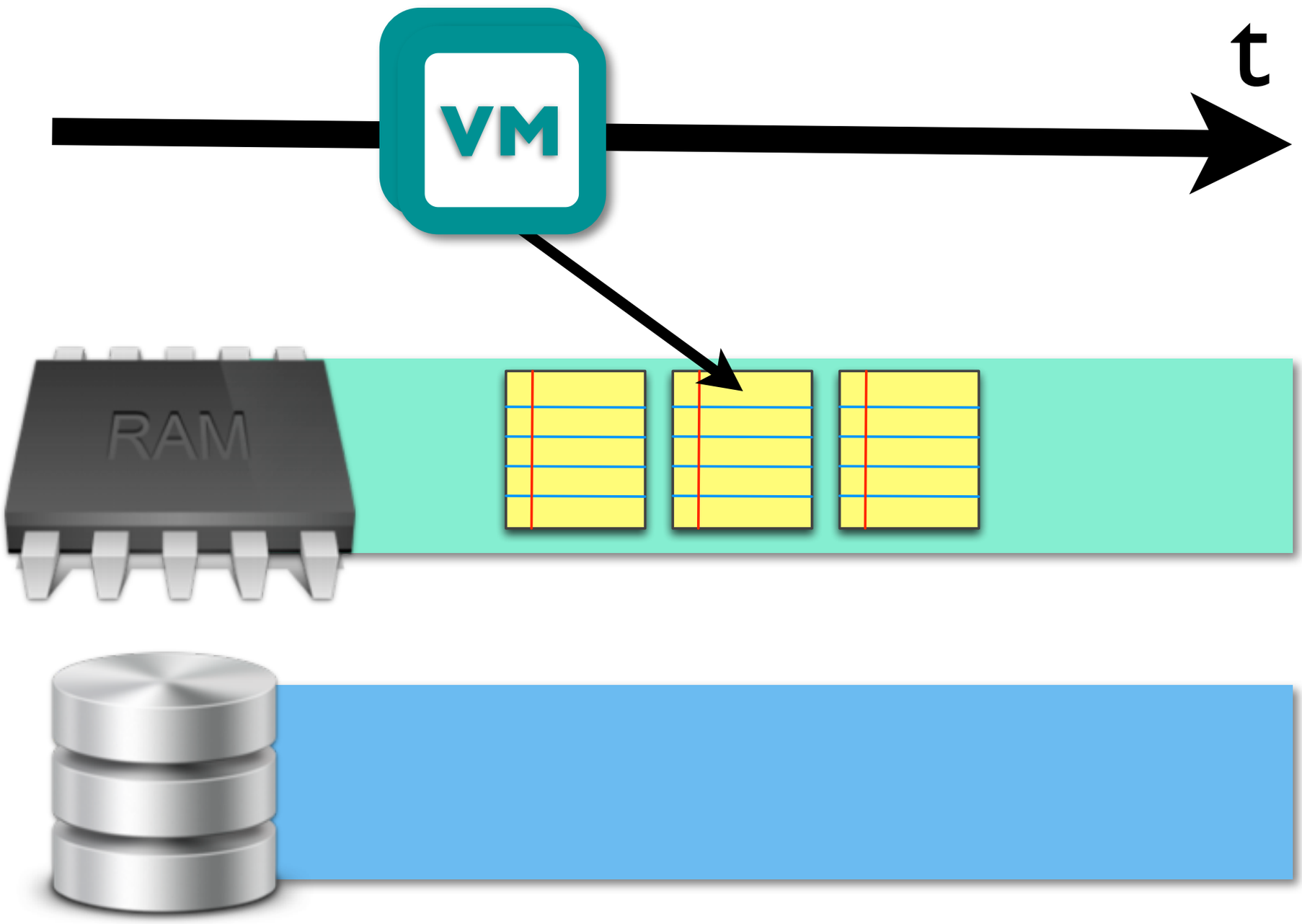
Halite Checkpoint Restore



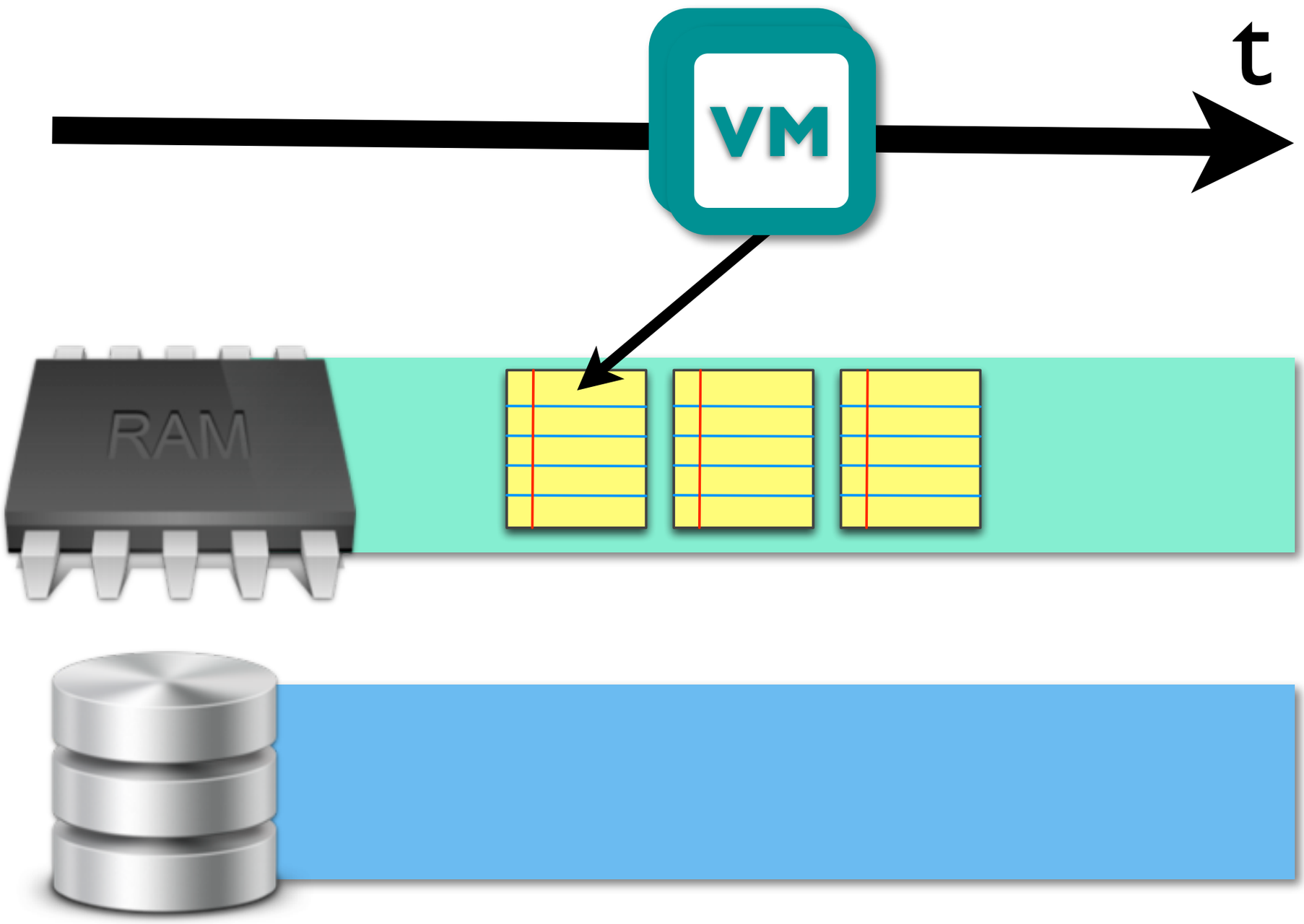
Halite Checkpoint Restore



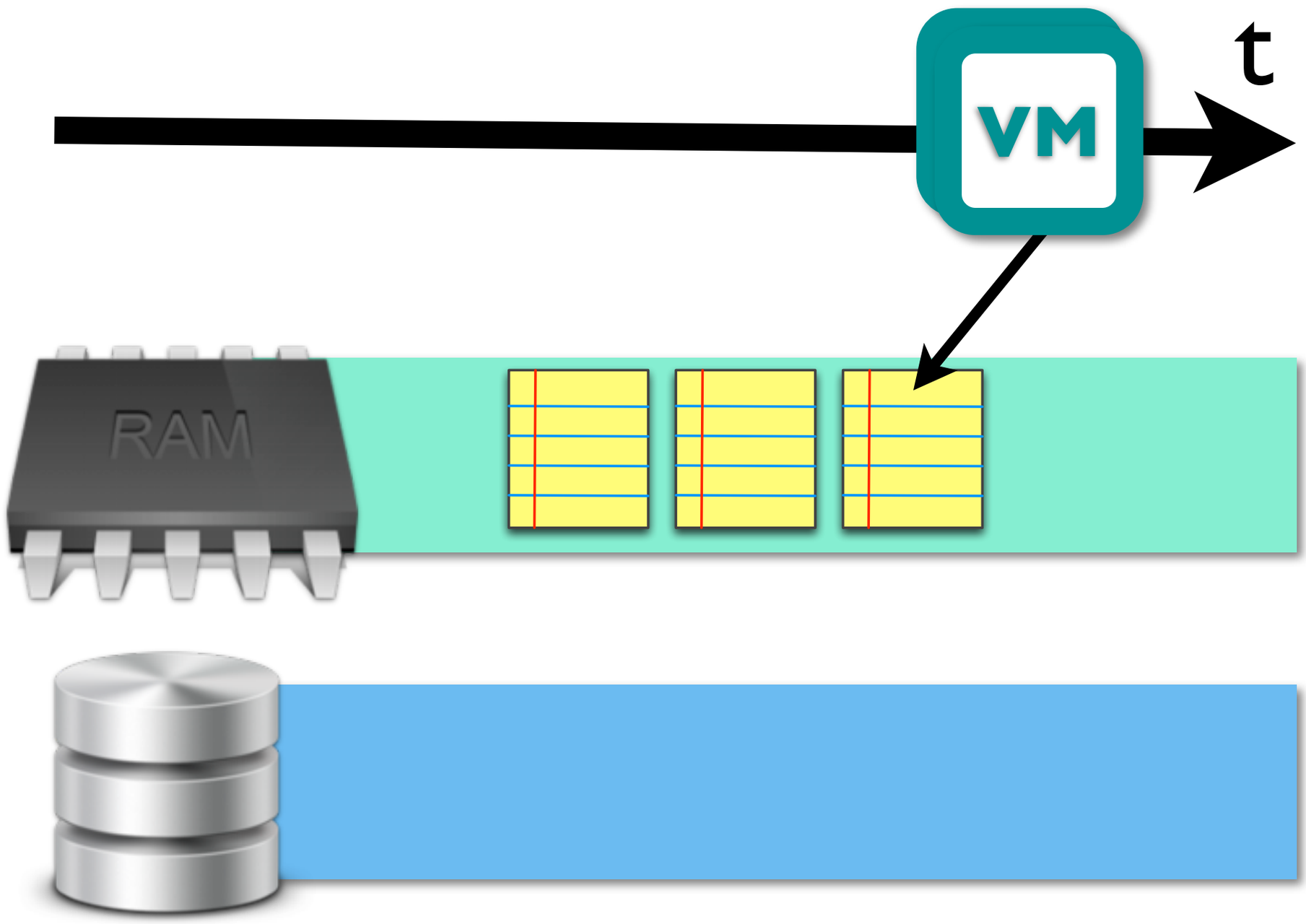
Halite Checkpoint Restore



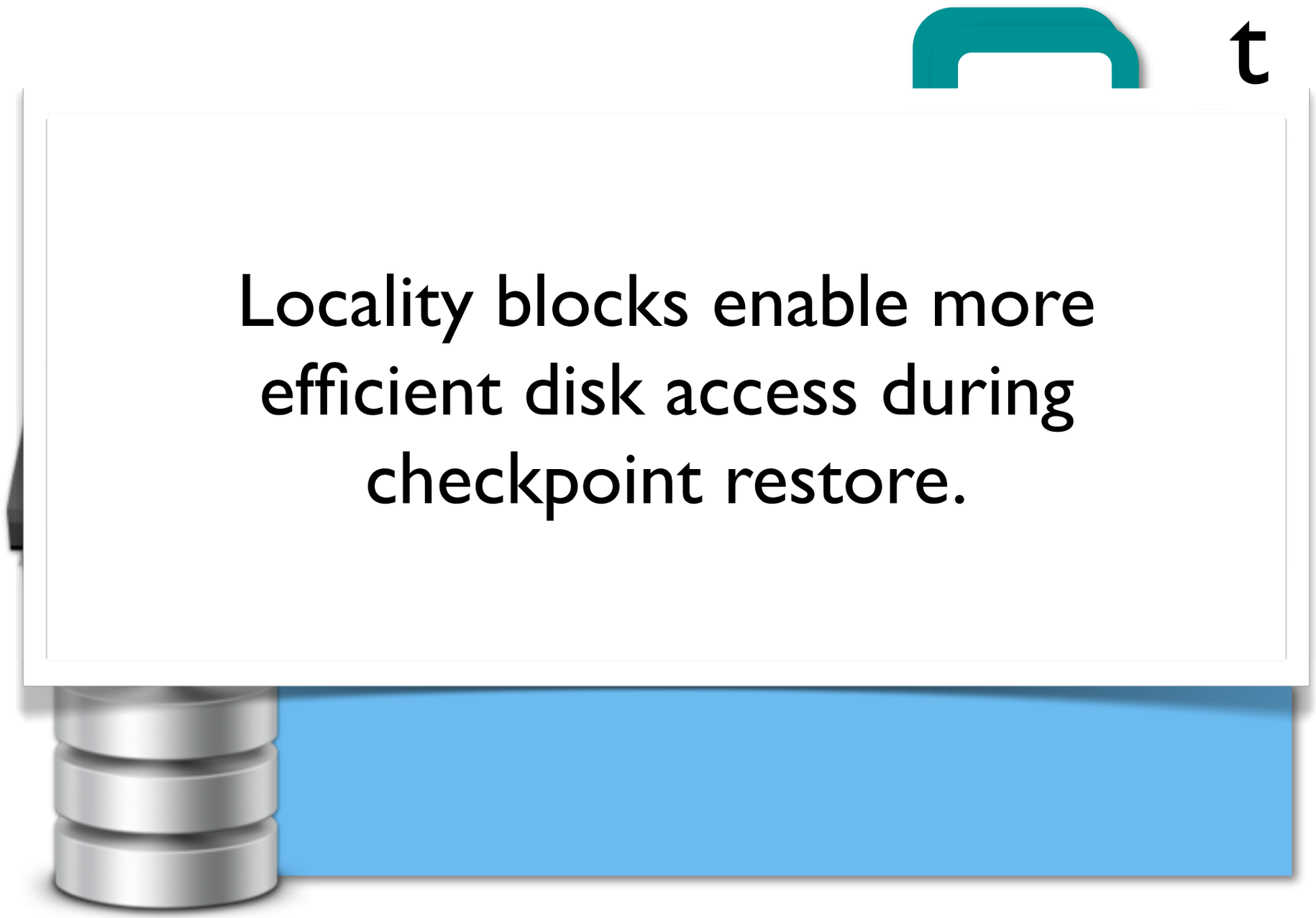
Halite Checkpoint Restore



Halite Checkpoint Restore



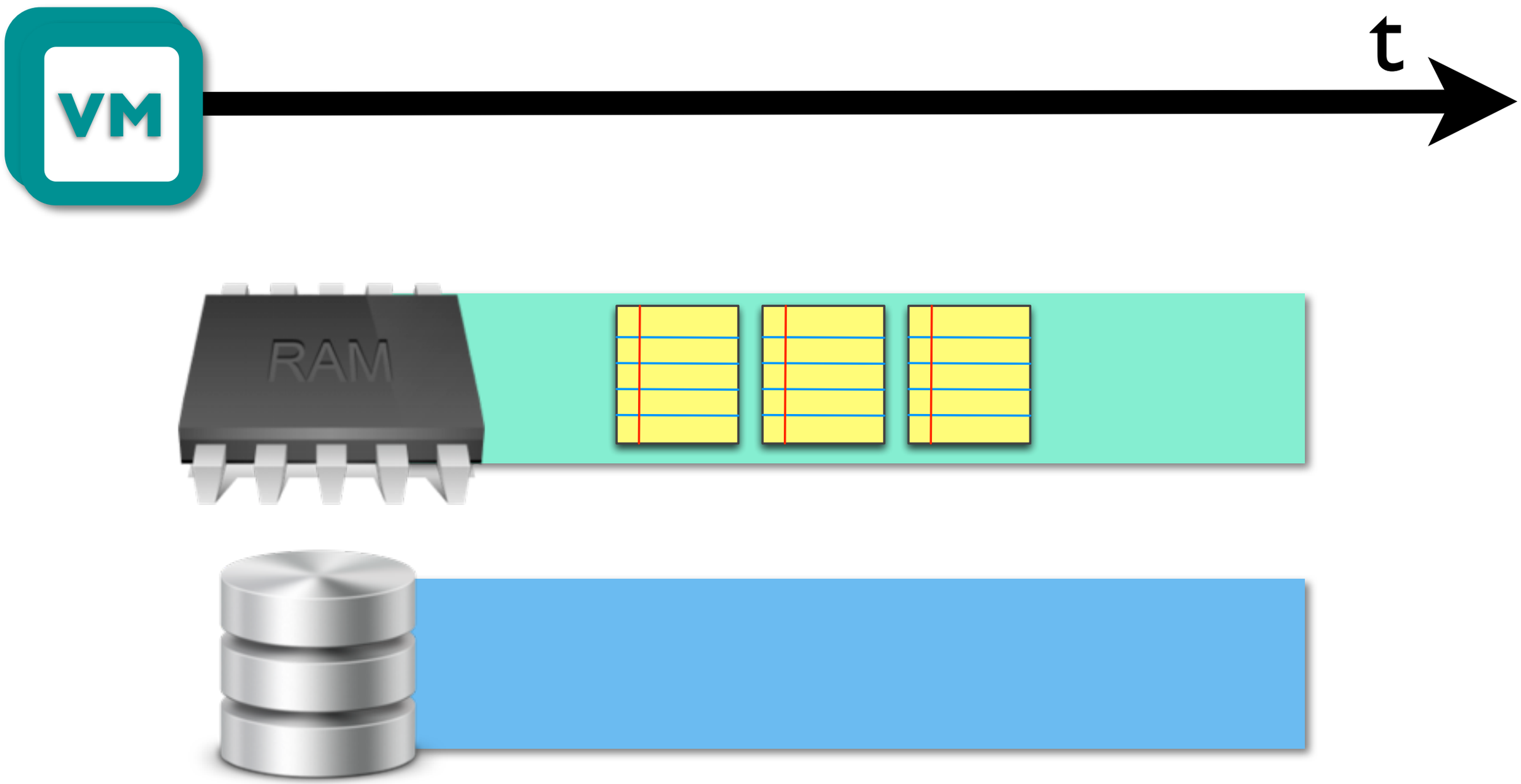
Halite Checkpoint Restore



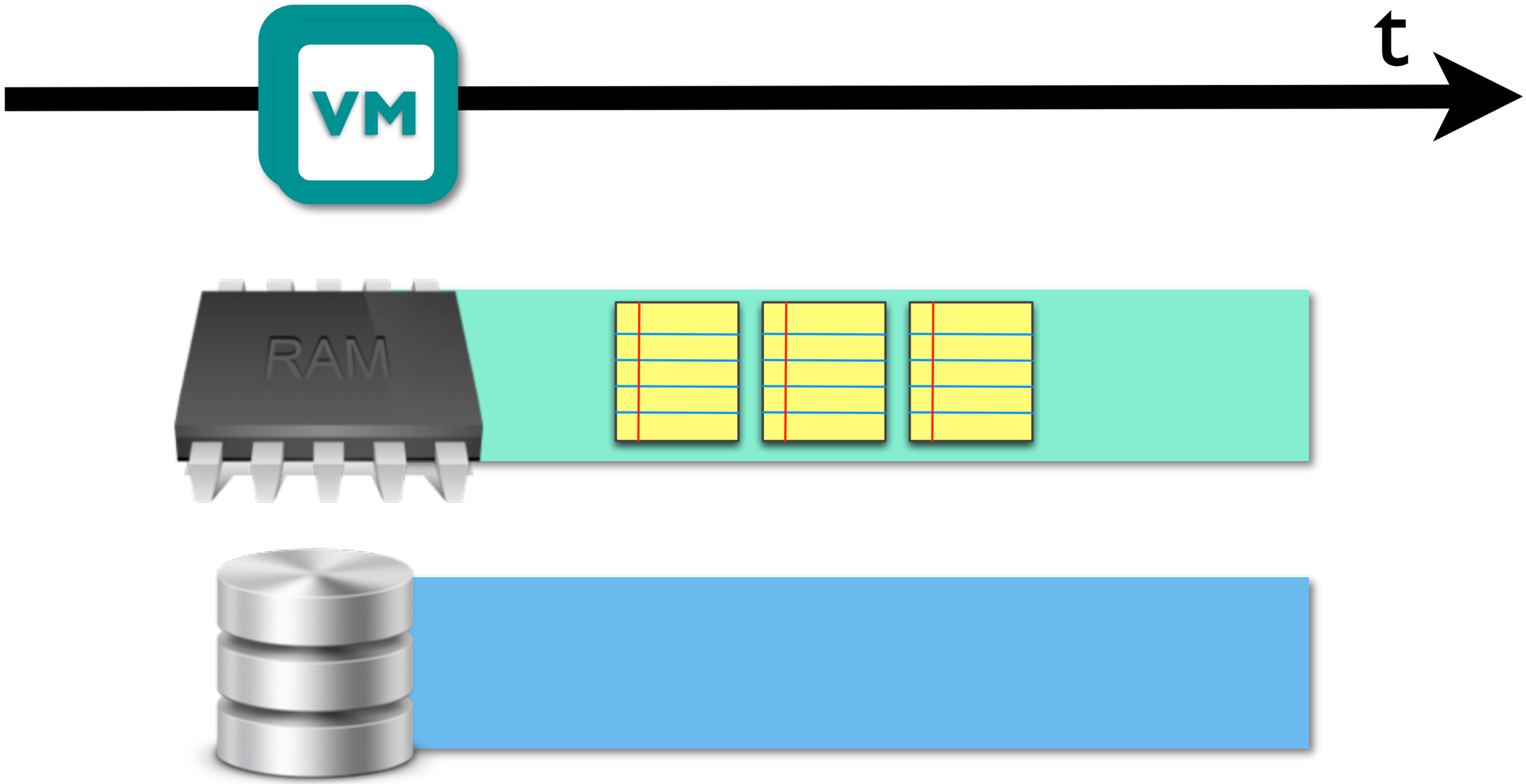
Access Locality Prediction

- Two techniques: checkpoint save access tracking and virtual address locality.
- Done entirely during checkpoint save.
- Directly used to save memory pages into locality blocks.

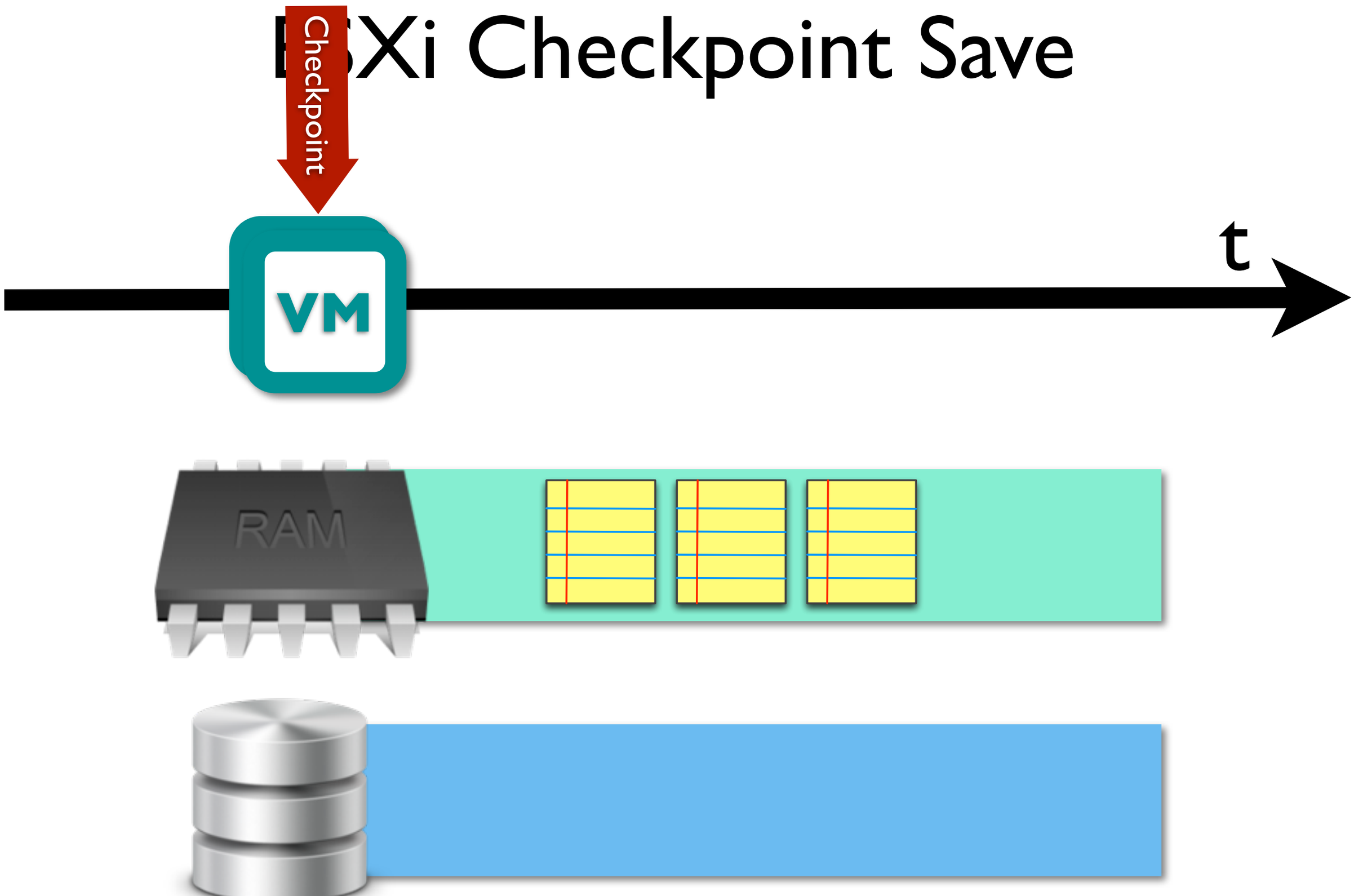
ESXi Checkpoint Save



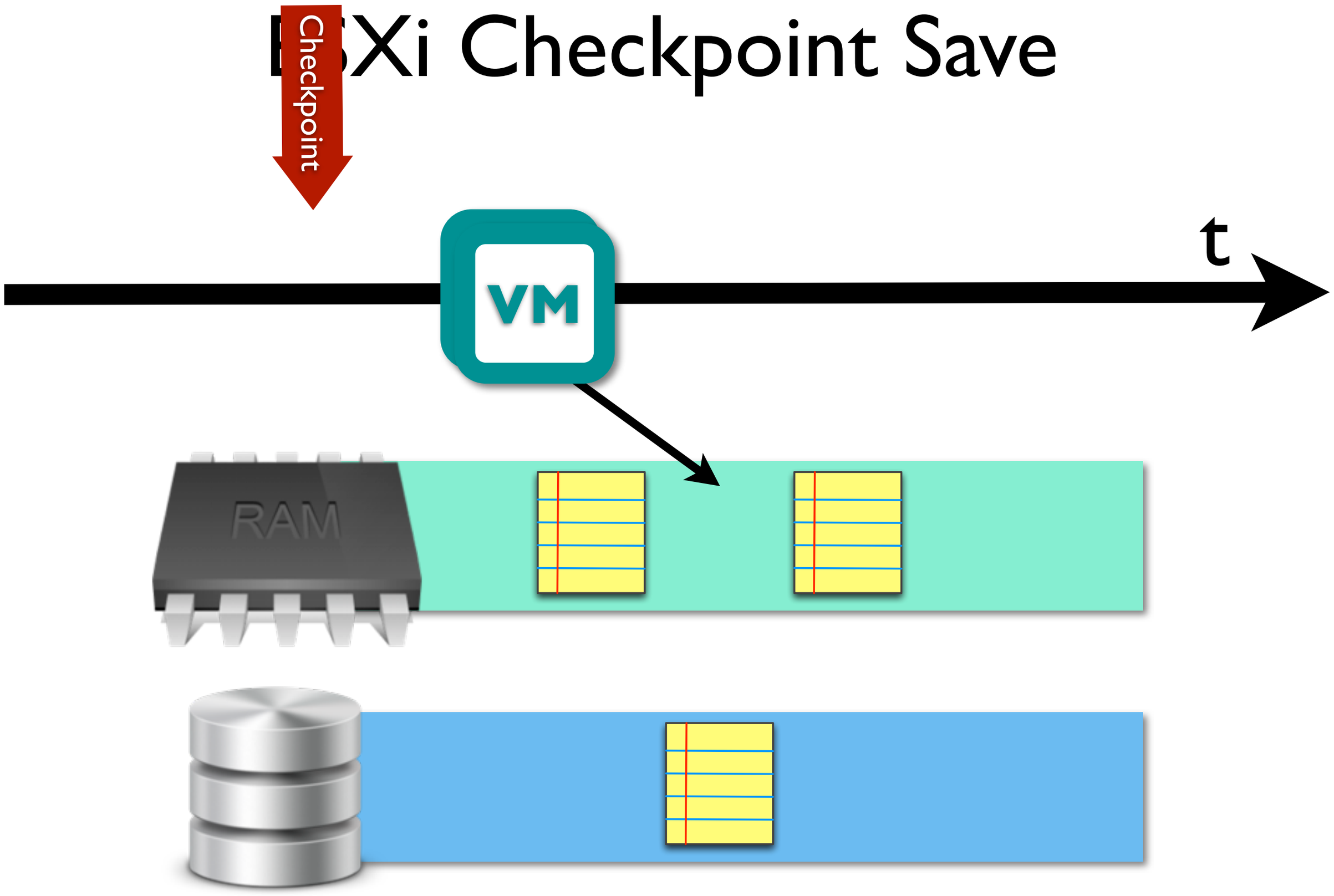
ESXi Checkpoint Save



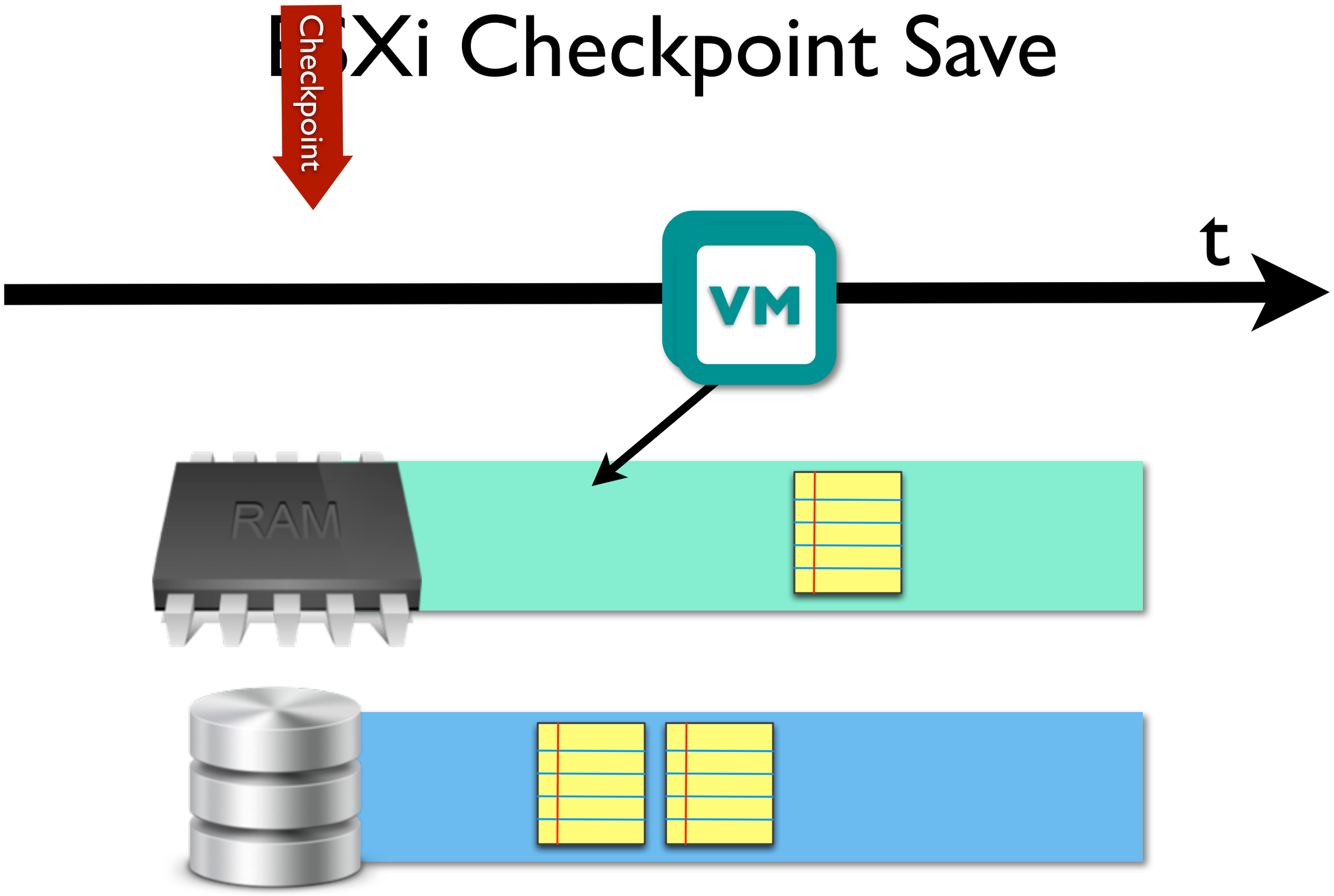
ESXi Checkpoint Save



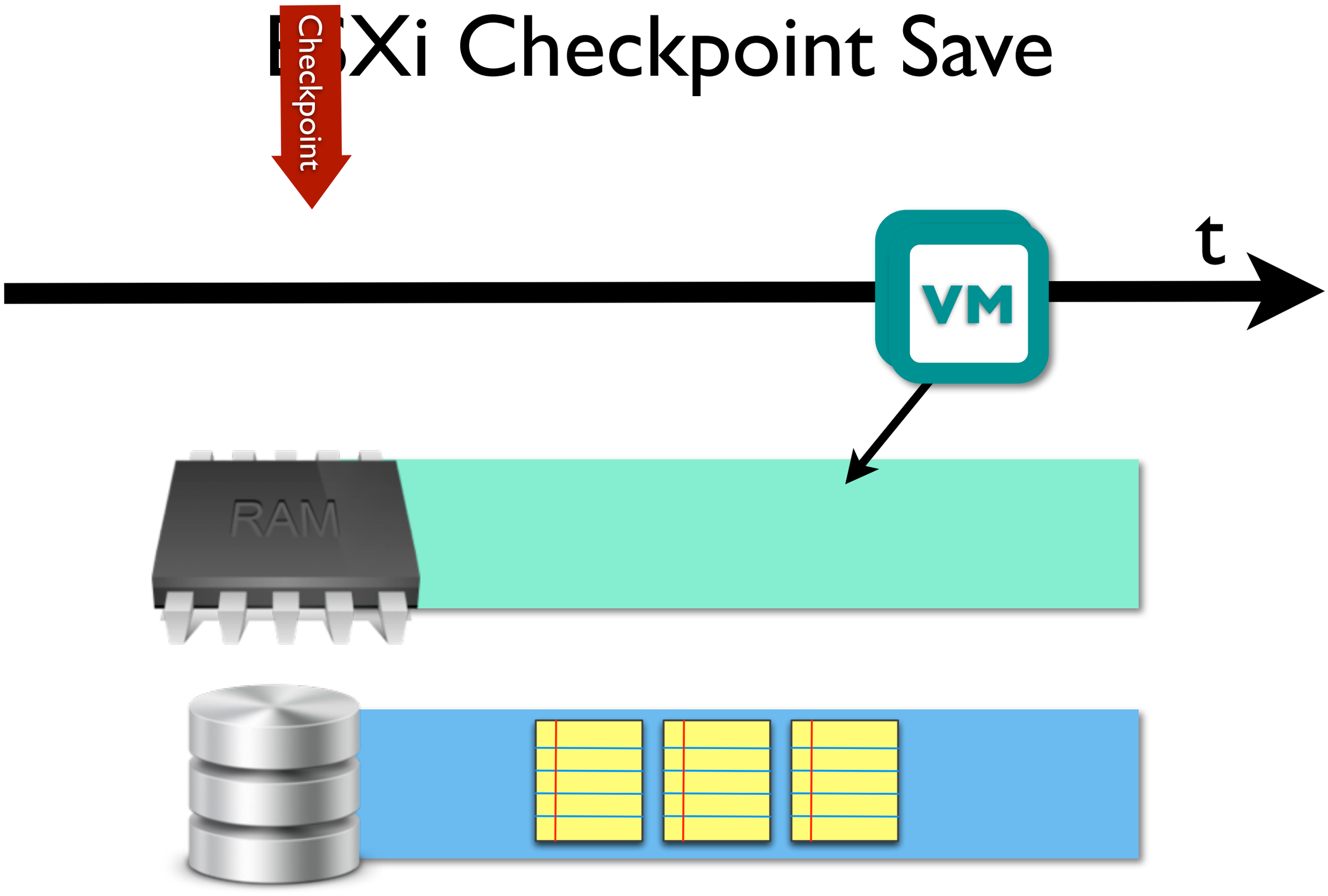
ESXi Checkpoint Save



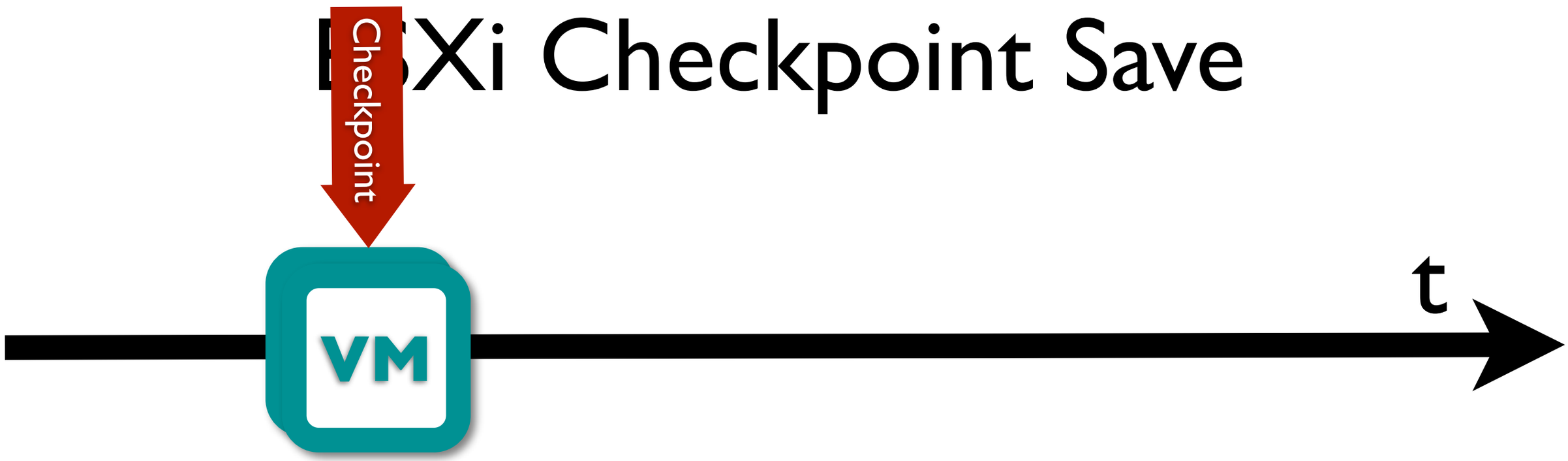
ESXi Checkpoint Save



ESXi Checkpoint Save



ESXi Checkpoint Save

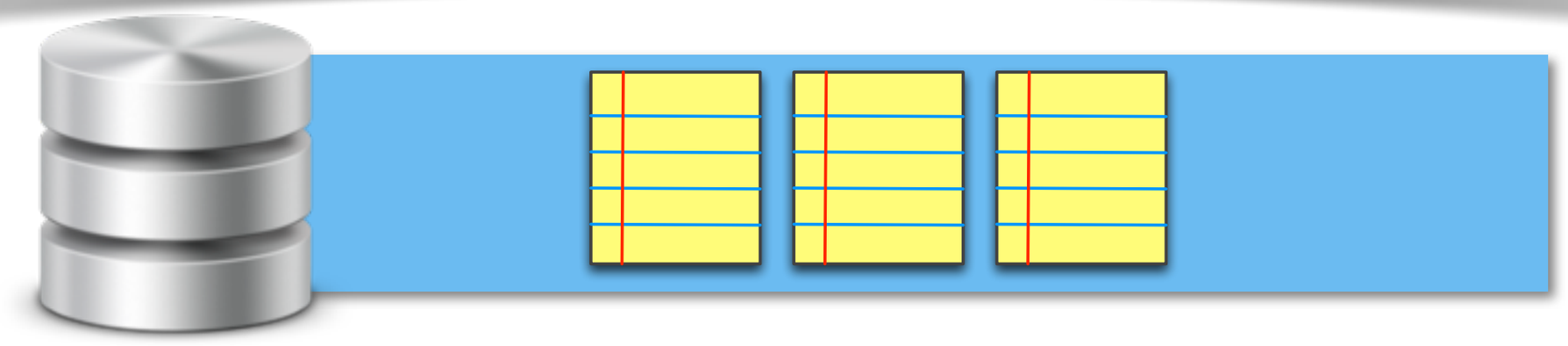


ESXi Checkpoint Save

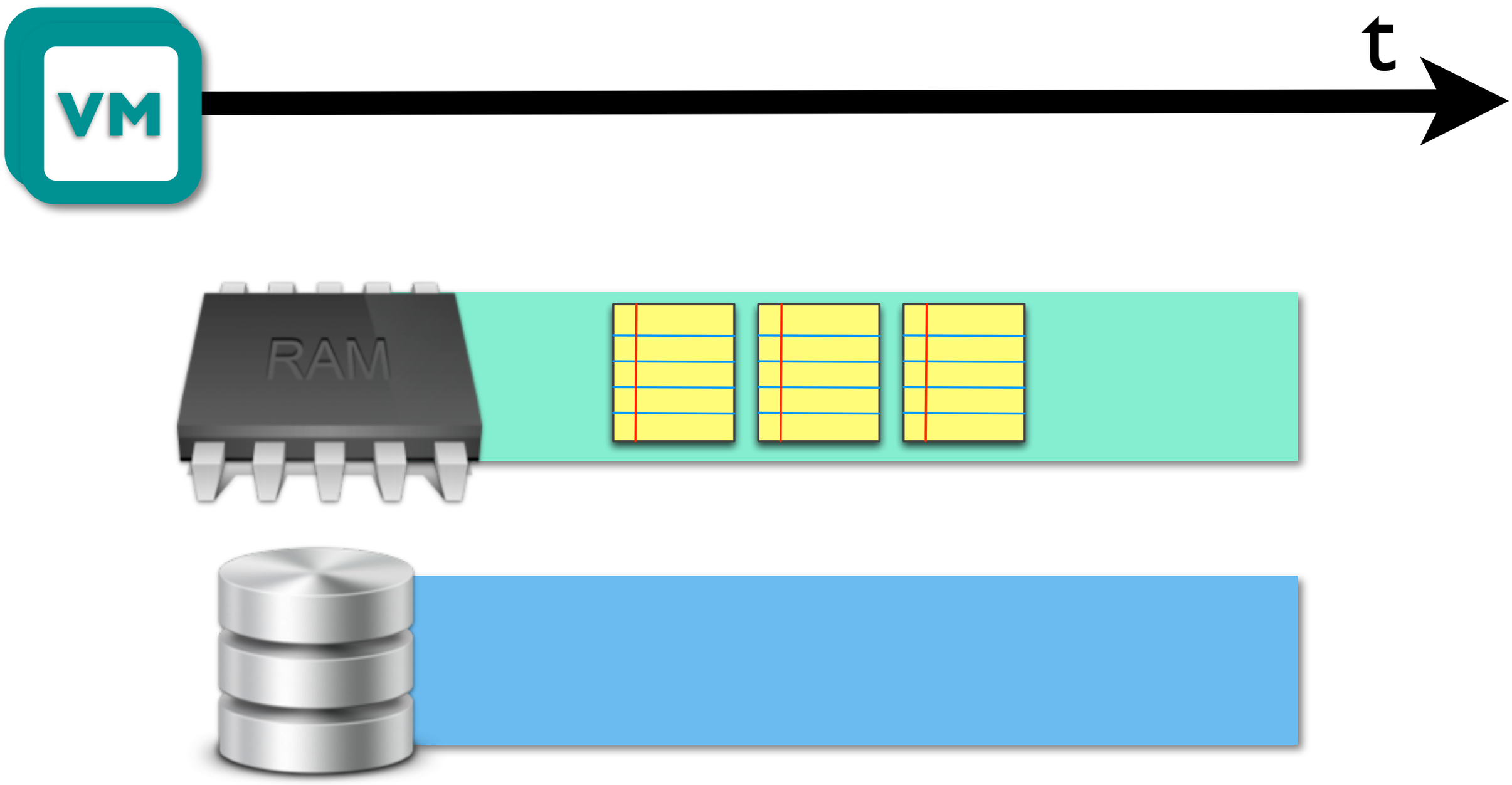
Checkpoi

t

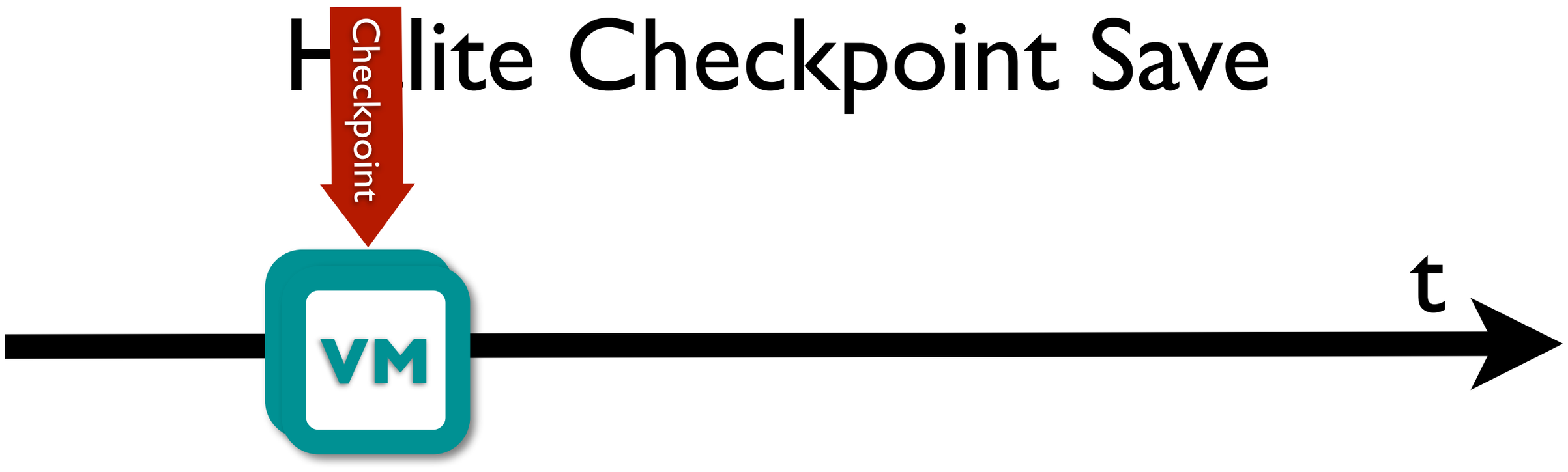
Pages accessed together during checkpoint save are likely to be accessed together again during checkpoint restore.

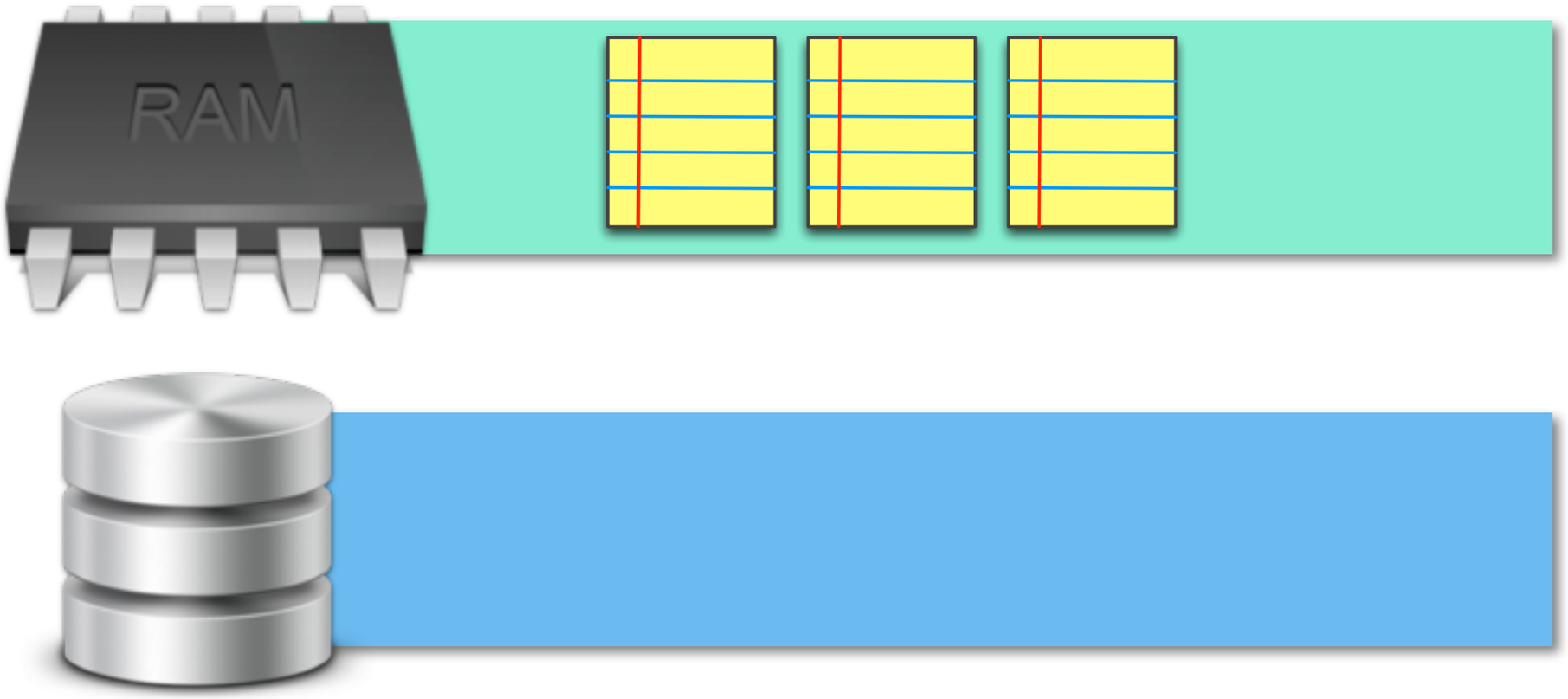


Halite Checkpoint Save

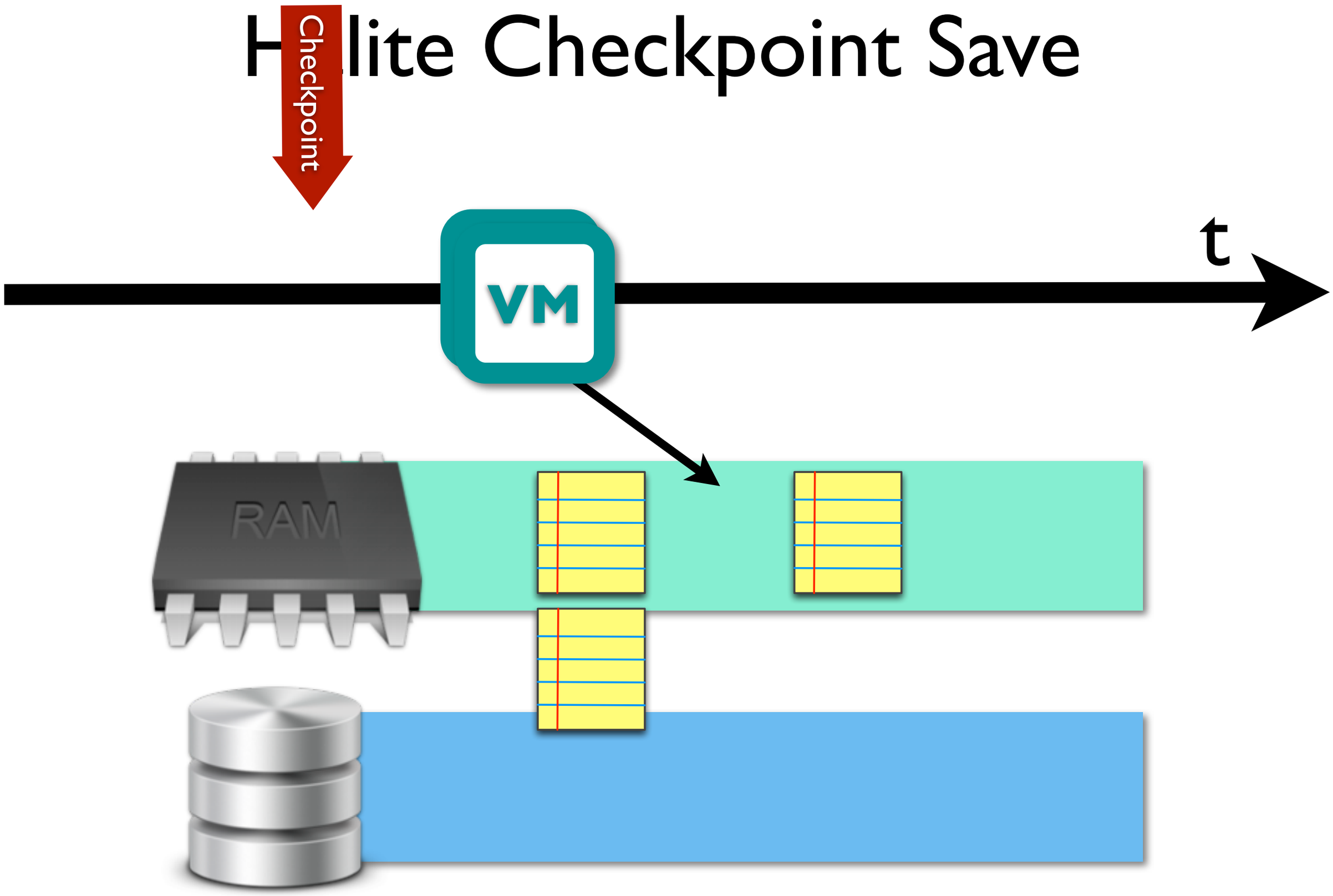


Hilite Checkpoint Save

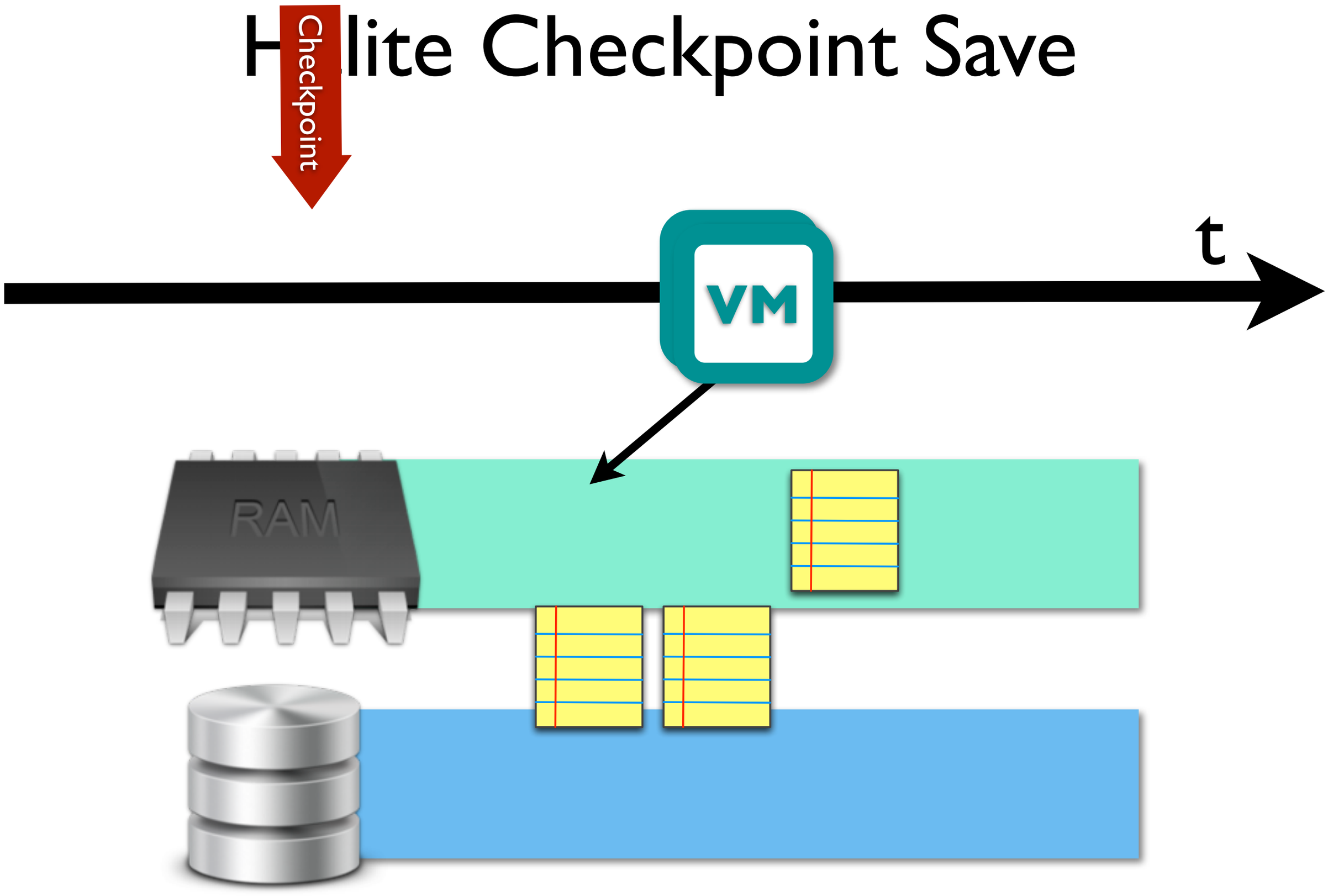




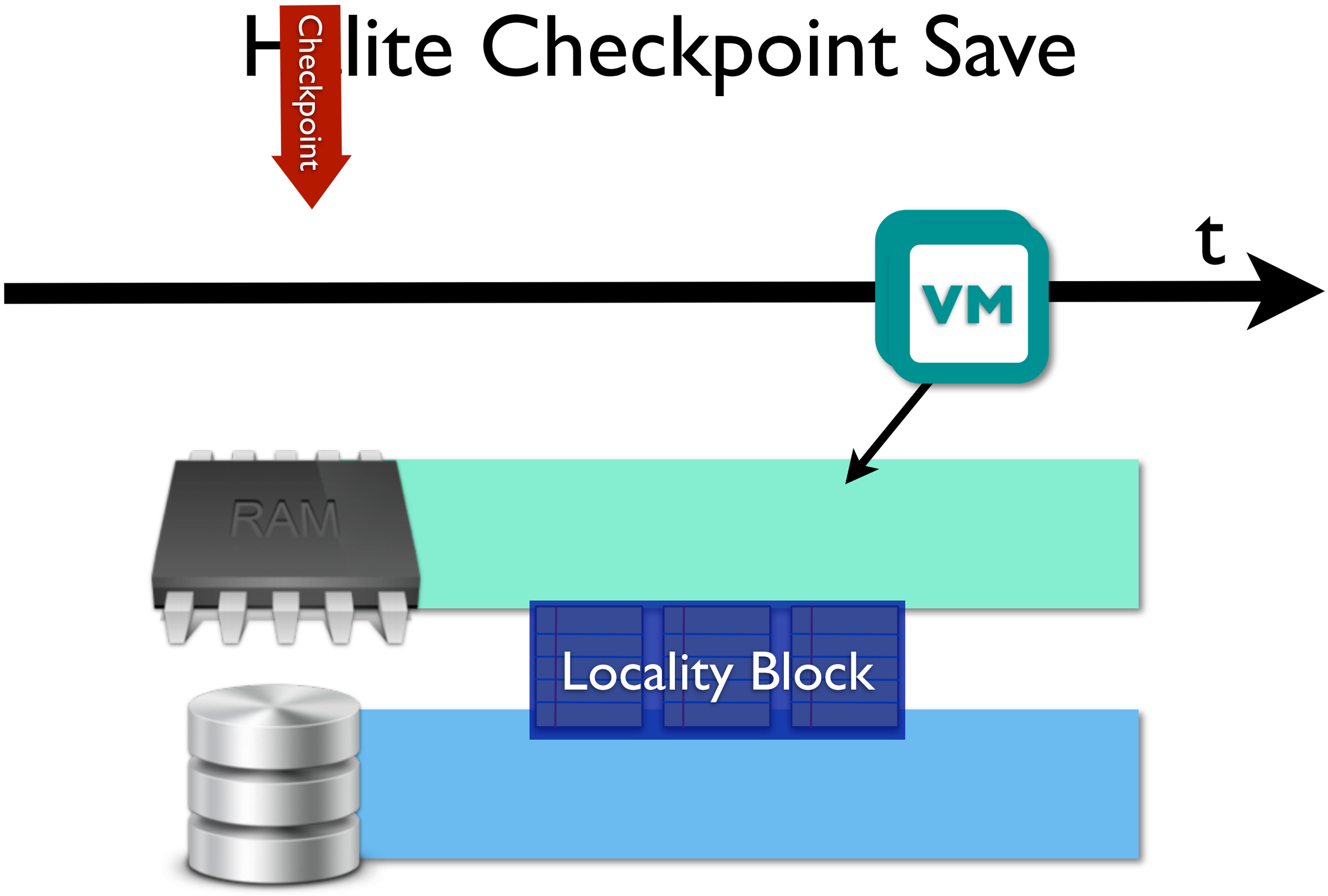
Hlite Checkpoint Save



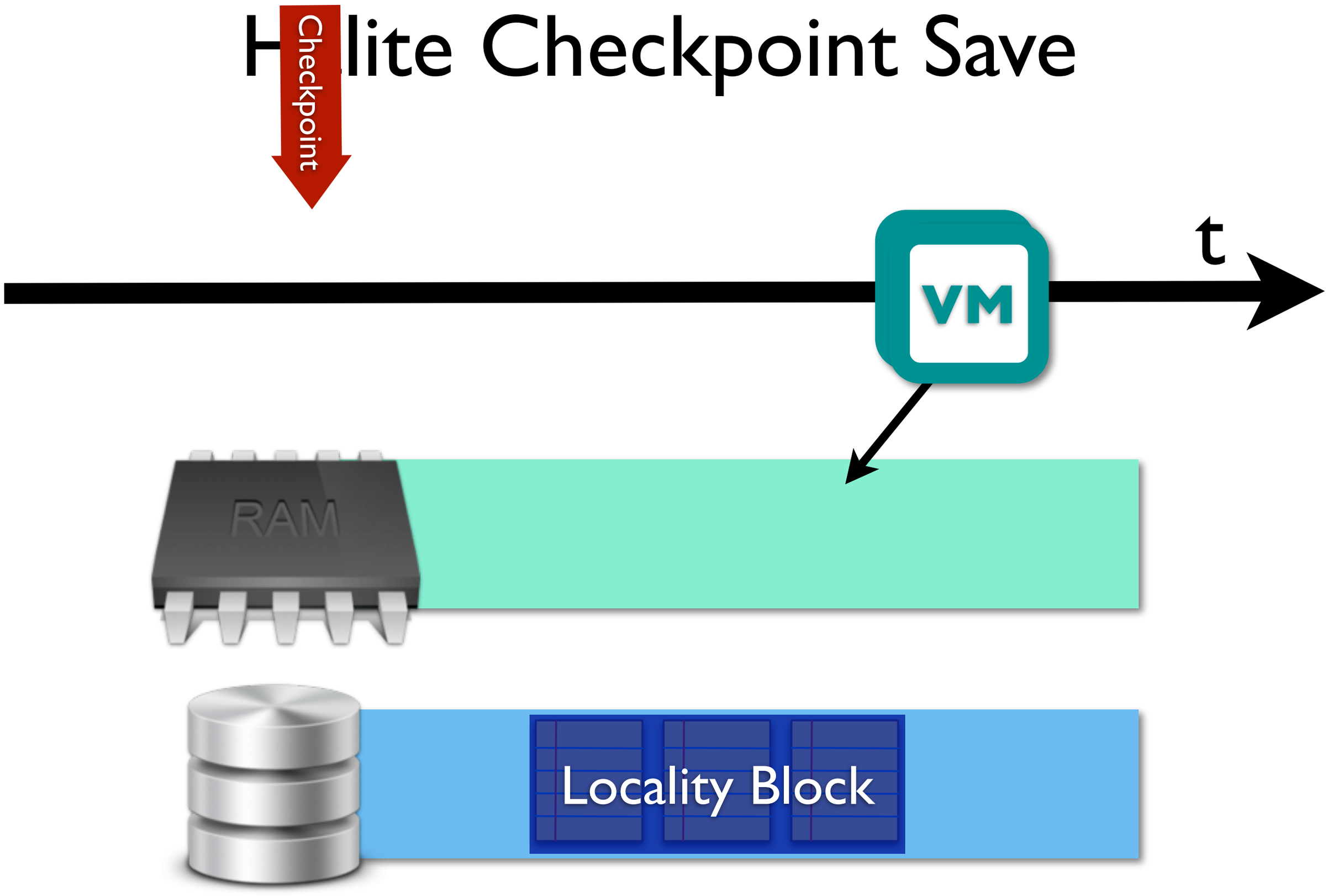
Hlite Checkpoint Save



Hlite Checkpoint Save



Hlite Checkpoint Save



Virtual Address Space Locality

- Used for pages the VM does not touch during checkpoint save.
- Pages mapped together in the guest virtual address space are likely to be accessed together.
- Background thread saves pages into locality blocks in virtual address order.

Other optimizations

- Compression
- Threading
- Background thread throttling
- Zero page optimization

Other optimizations

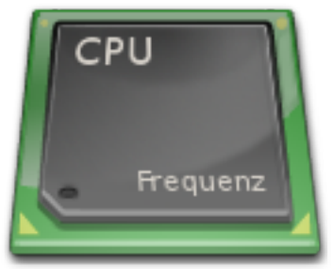
- Compression
- Threading
- Background thread throttling
- Zero page optimization

Please refer to the paper.

Evaluation

- Does Halite improve restore performance?
- How much do locality blocks improve performance?
- Could we use blocks without access locality?
- Does Halite affect checkpoint save performance?

Hardware



2.3 GHz 8-core AMD Opteron

24 GB

15K RPM Seagate ITB (VMFS-5)

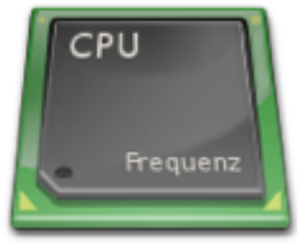
Workloads

sim	Custom memory access simulator
pgbench	Built-in PostgreSQL benchmark
worldbench	Windows desktop benchmark
apache webserver	HTTP get benchmark

Workloads

sim	Custom memory access simulator
pgbench	Built-in PostgreSQL benchmark
worldbench	Windows desktop benchmark
apache webserver	HTTP get benchmark

VM Configuration

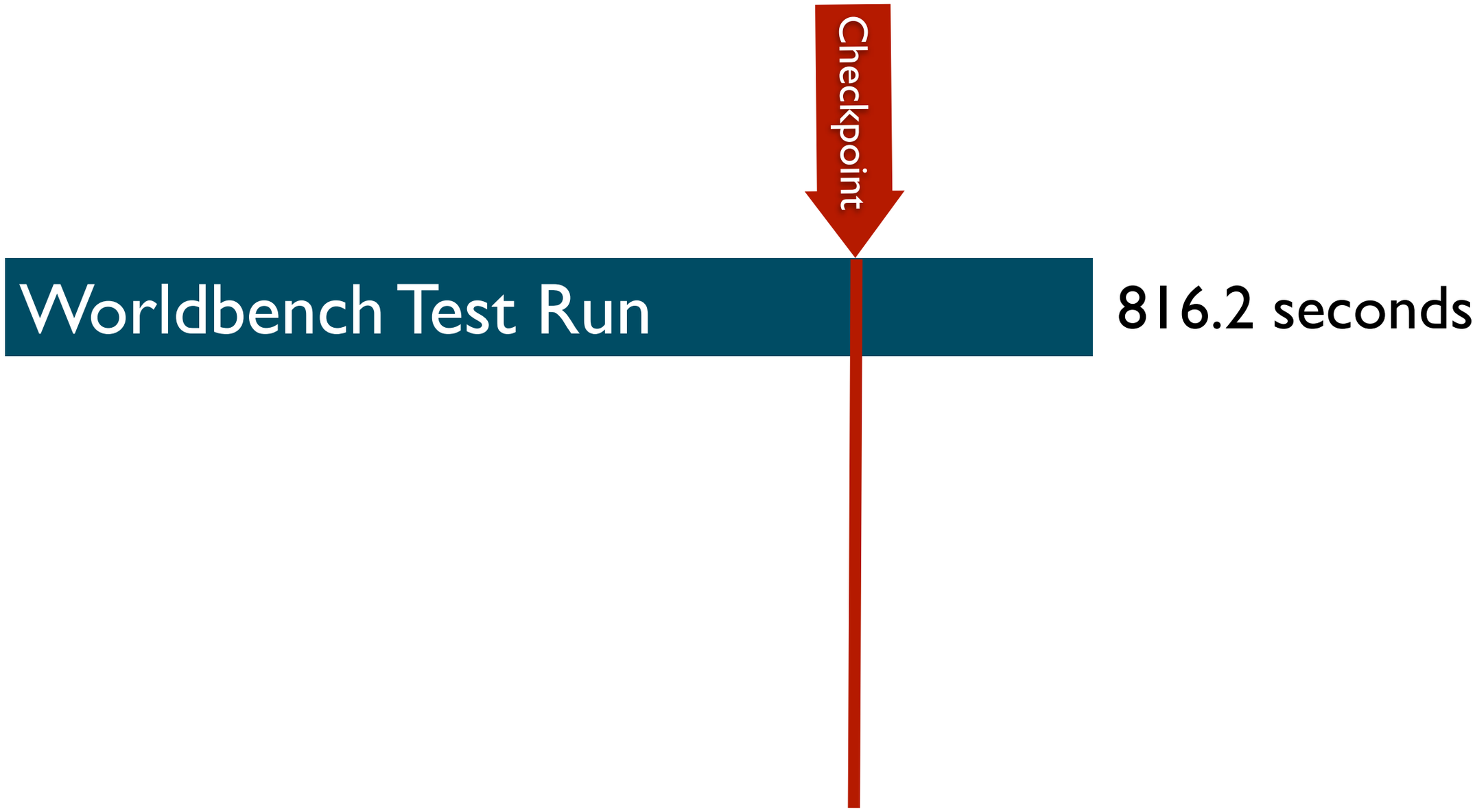


sim	4 vCPU	2 GB	
pgbench	4 vCPU	2 GB	
worldbench	2 vCPU	1 GB	
apache webserver	2 vCPU	2 GB	

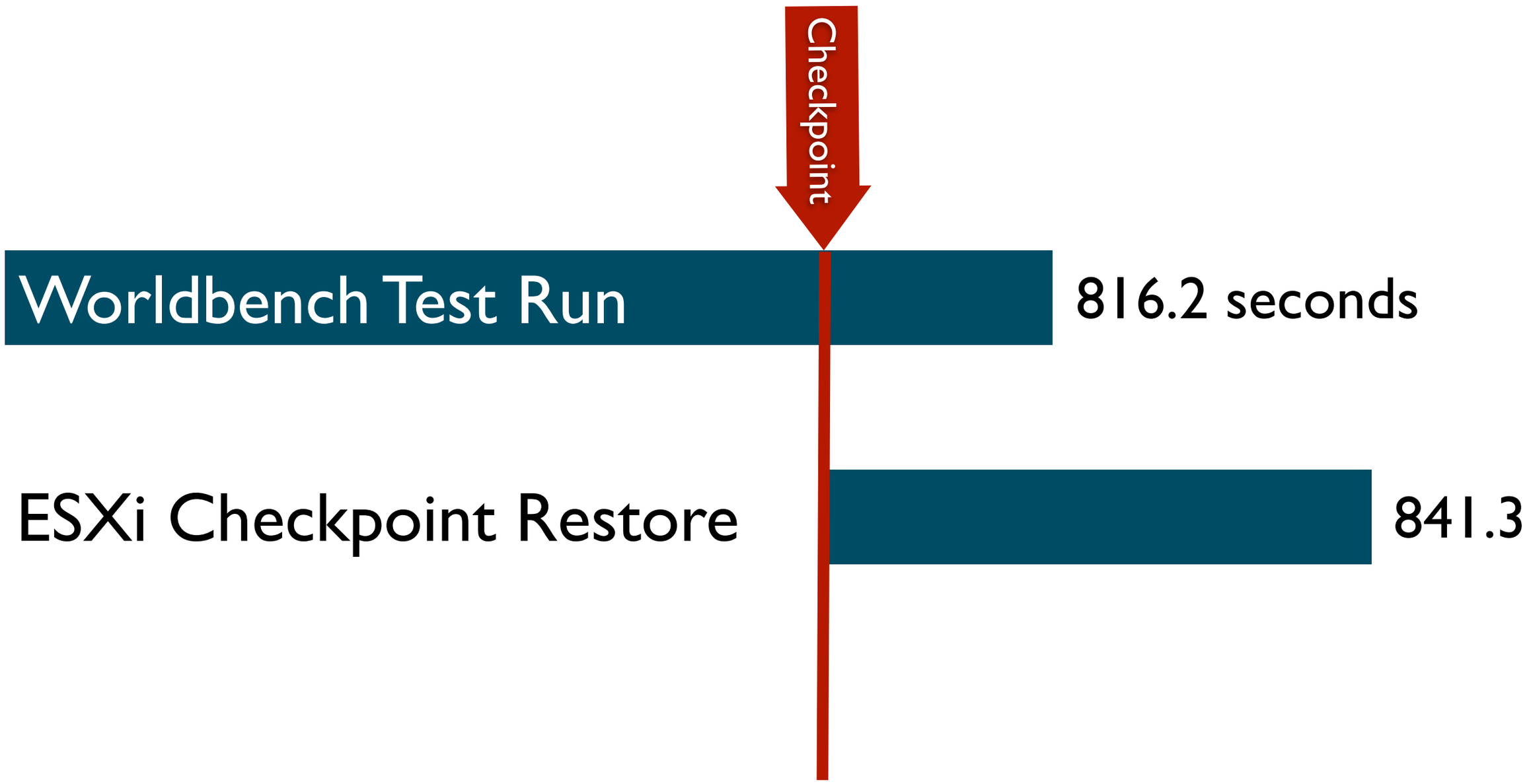
Worldbench Test Setup

Worldbench Test Run	816.2 seconds

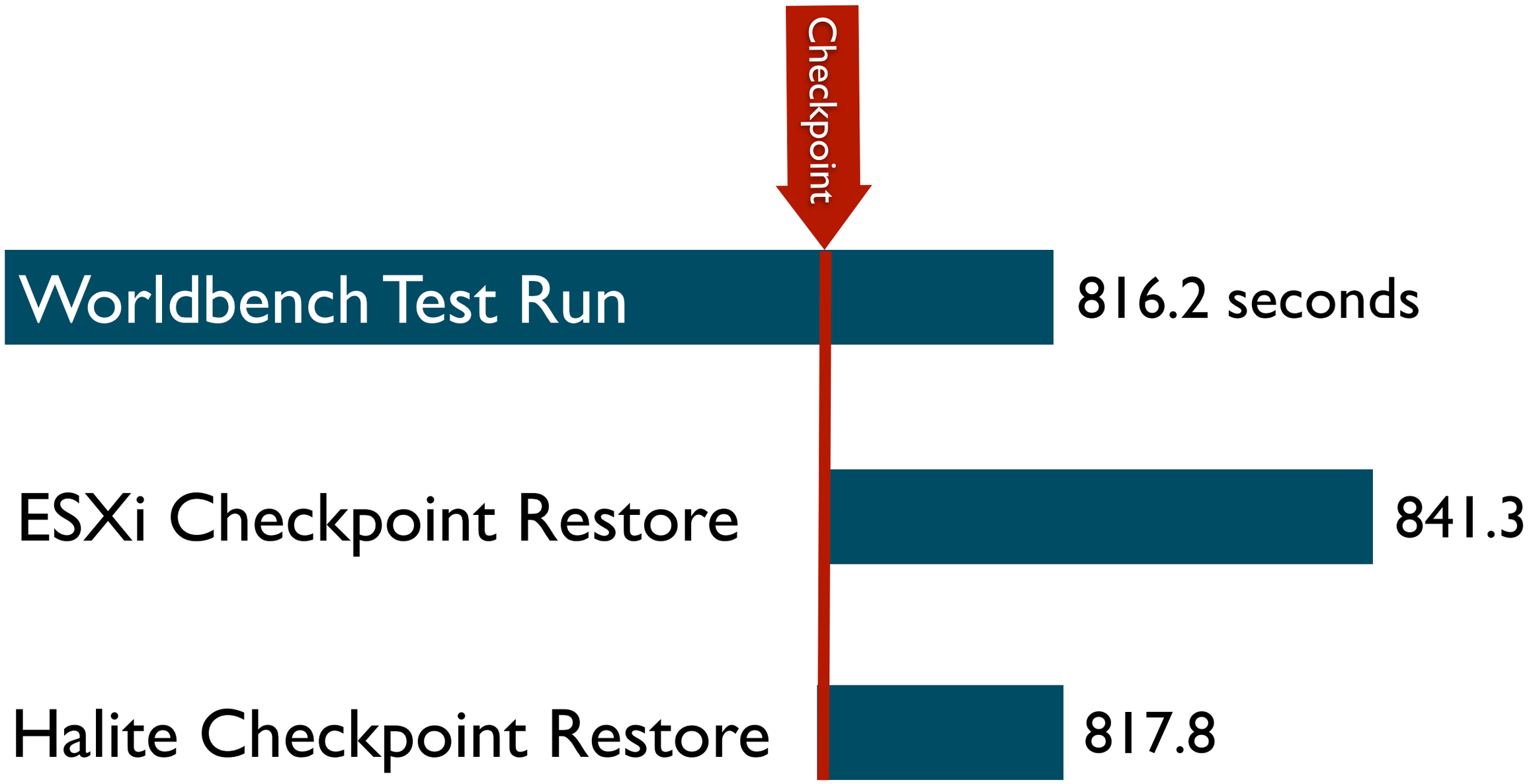
Worldbench Test Setup



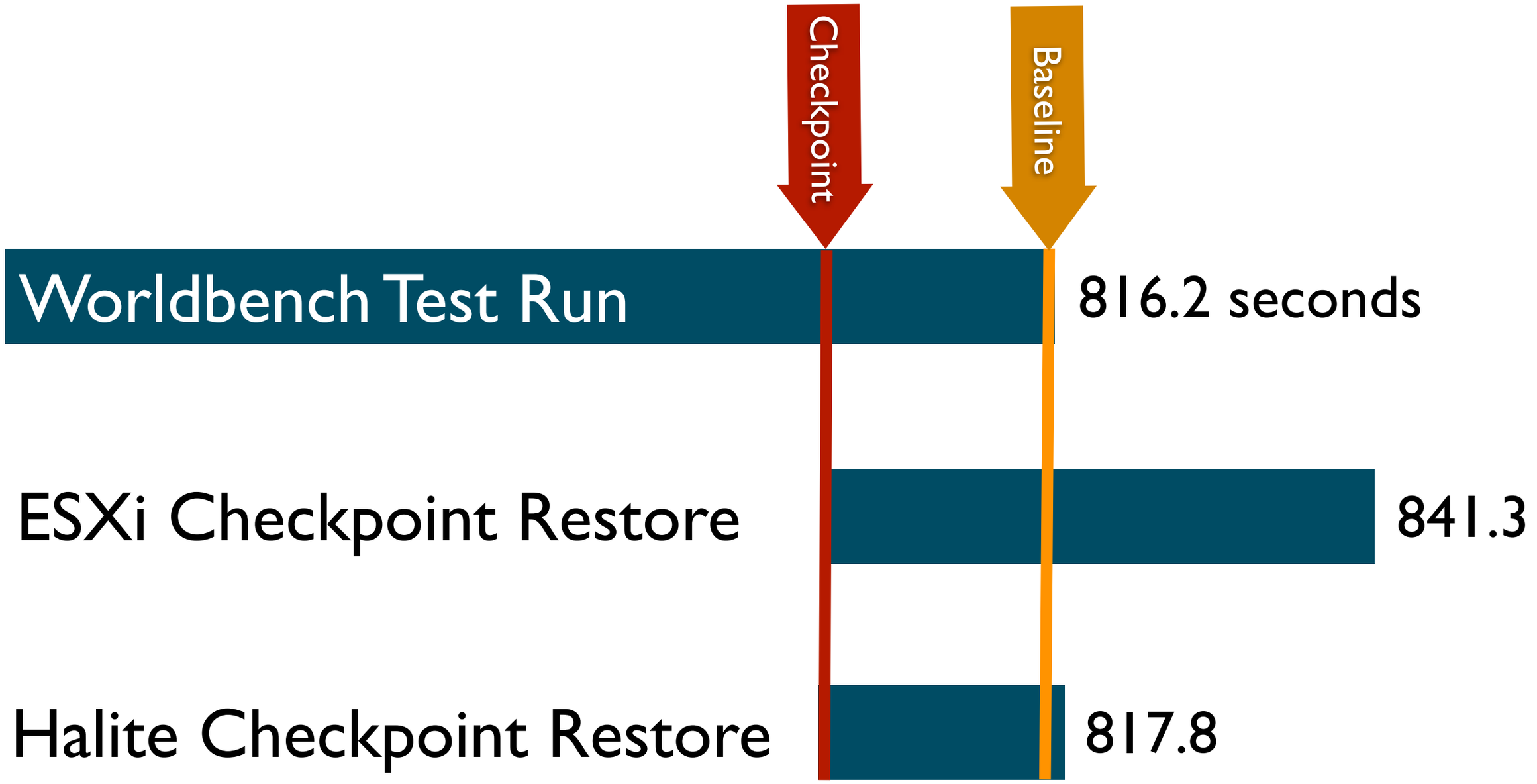
Worldbench Test Setup



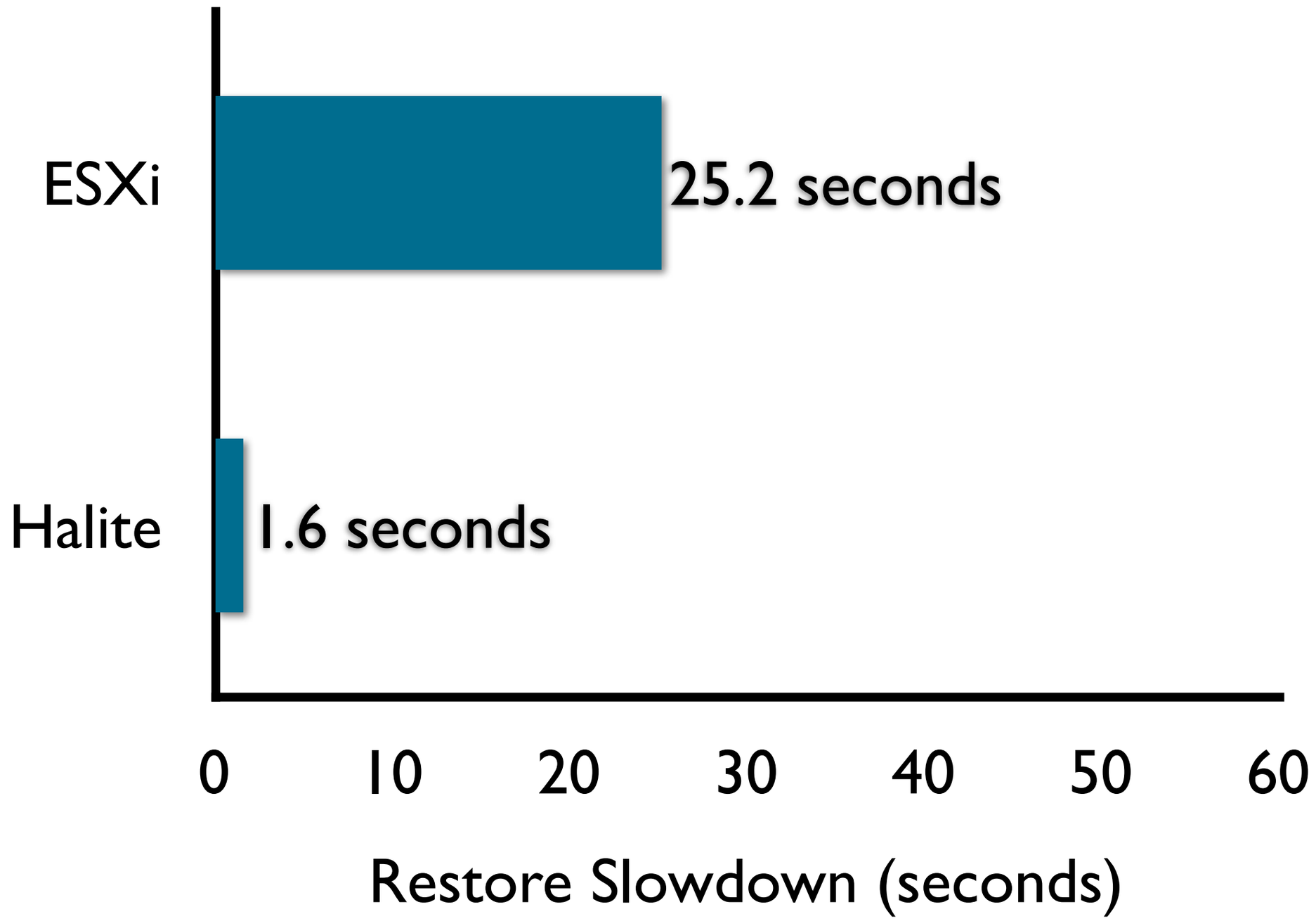
Worldbench Test Setup



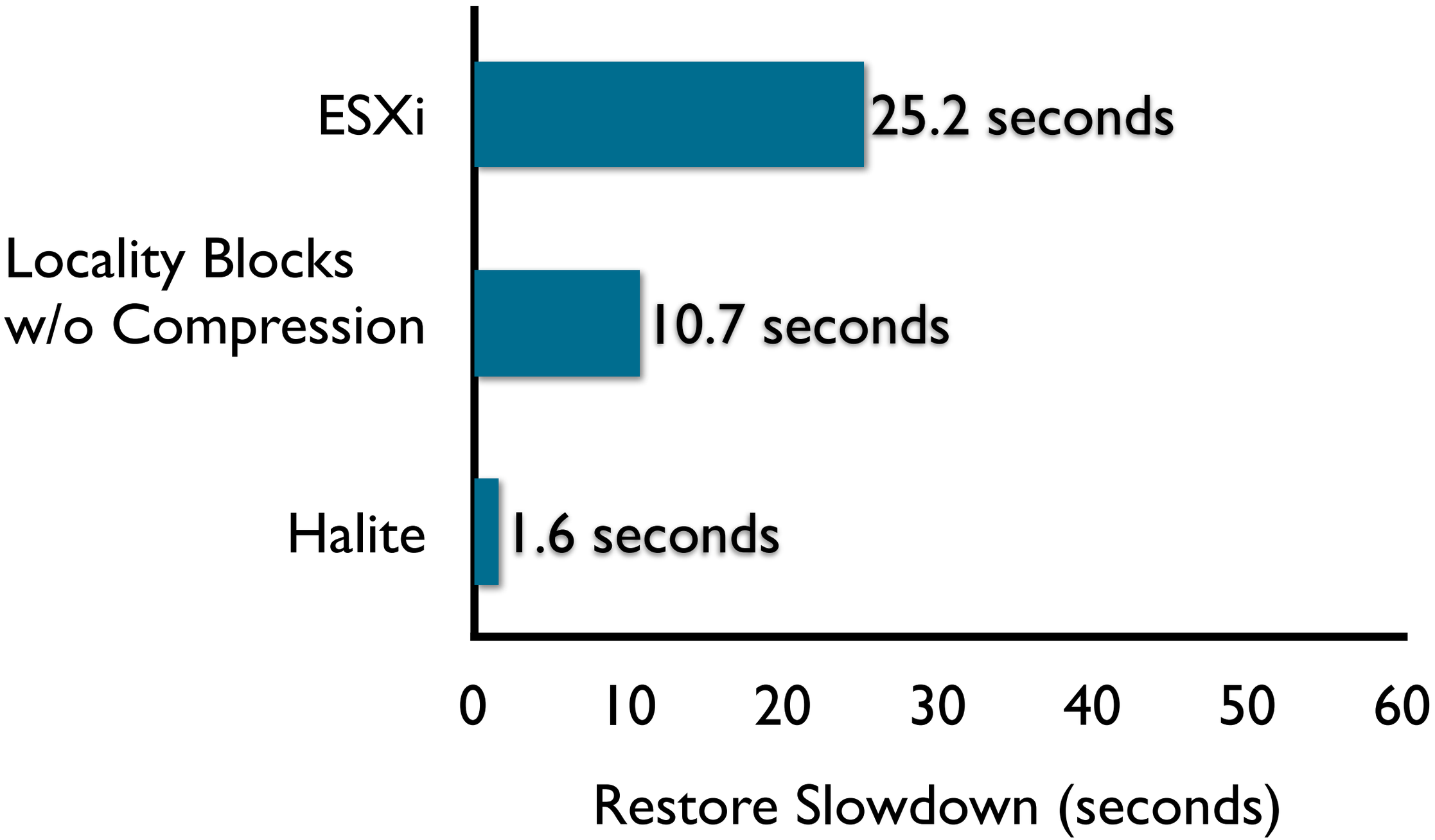
Worldbench Test Setup



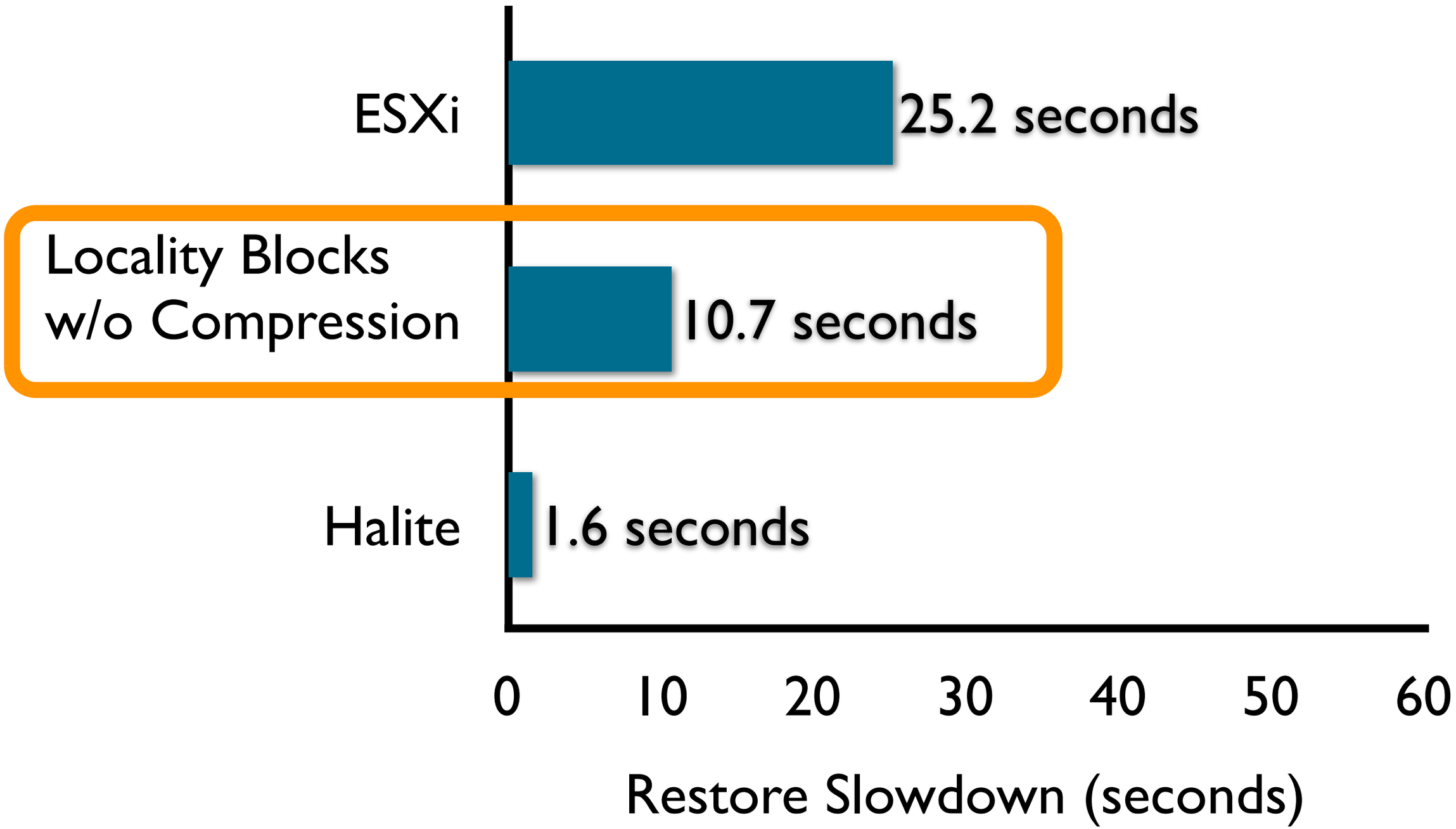
Halite reduces restore overhead from tens of seconds to seconds.



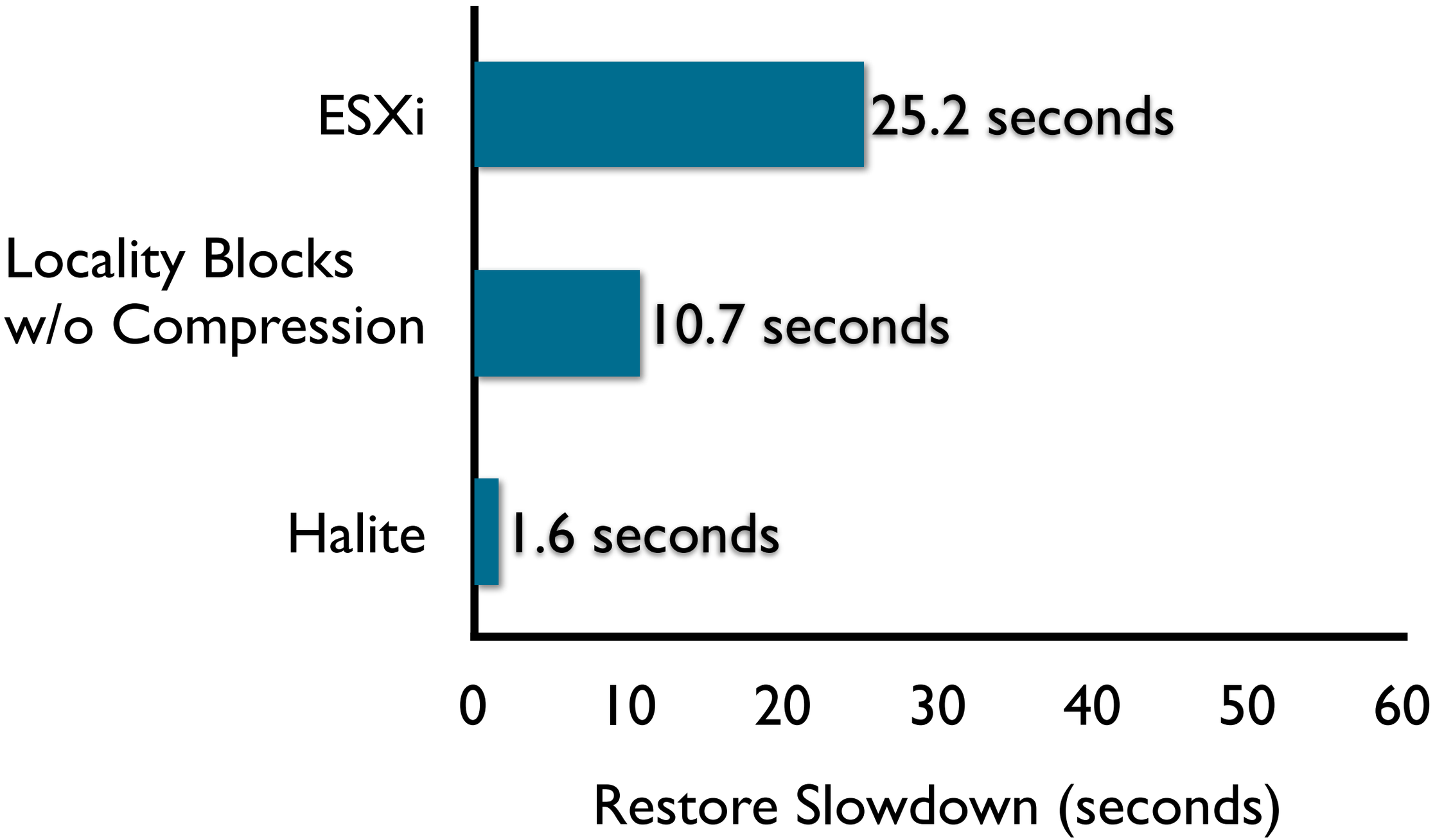
Locality blocks provide more than half of Halite's performance improvement.



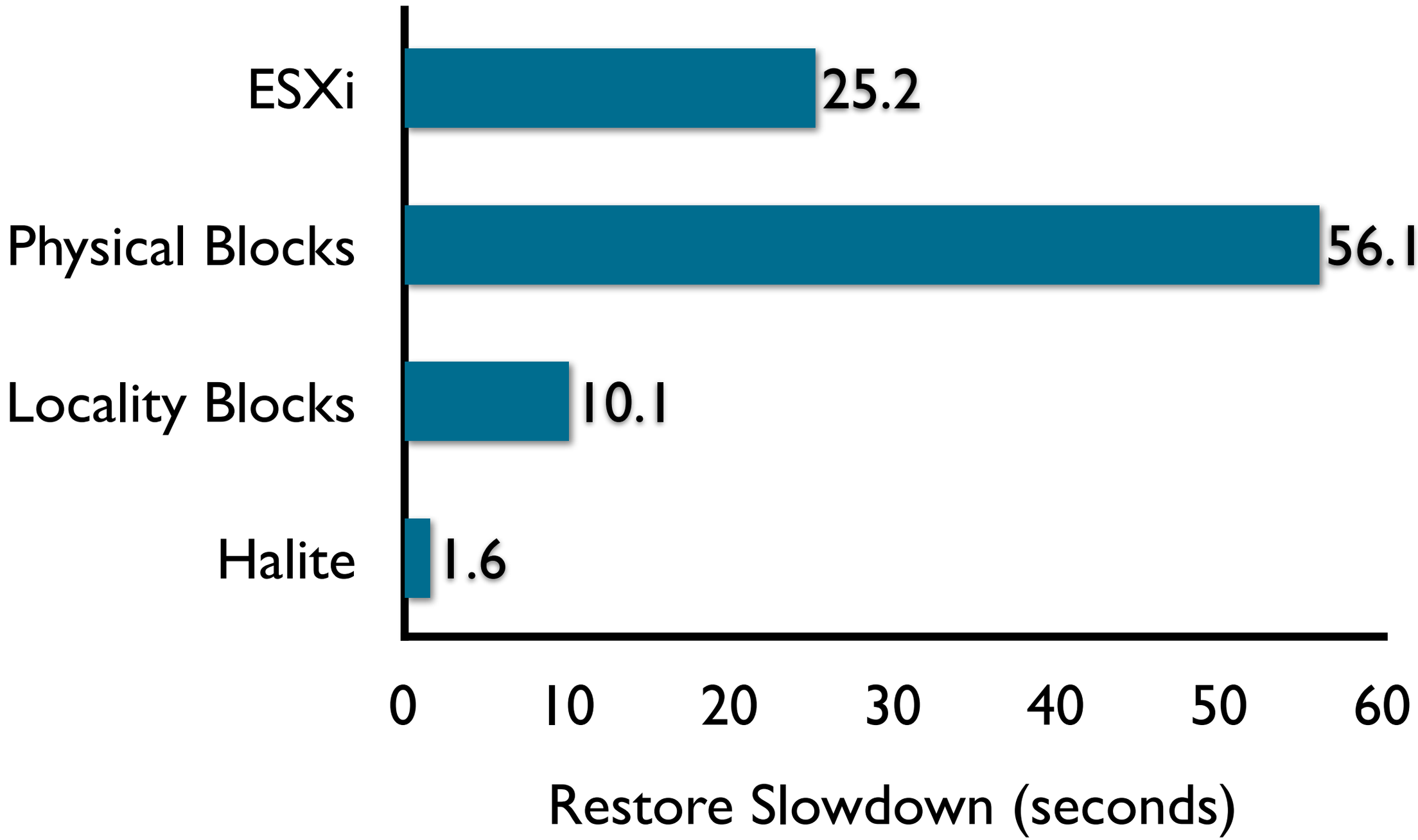
Locality blocks provide more than half of Halite's performance improvement.



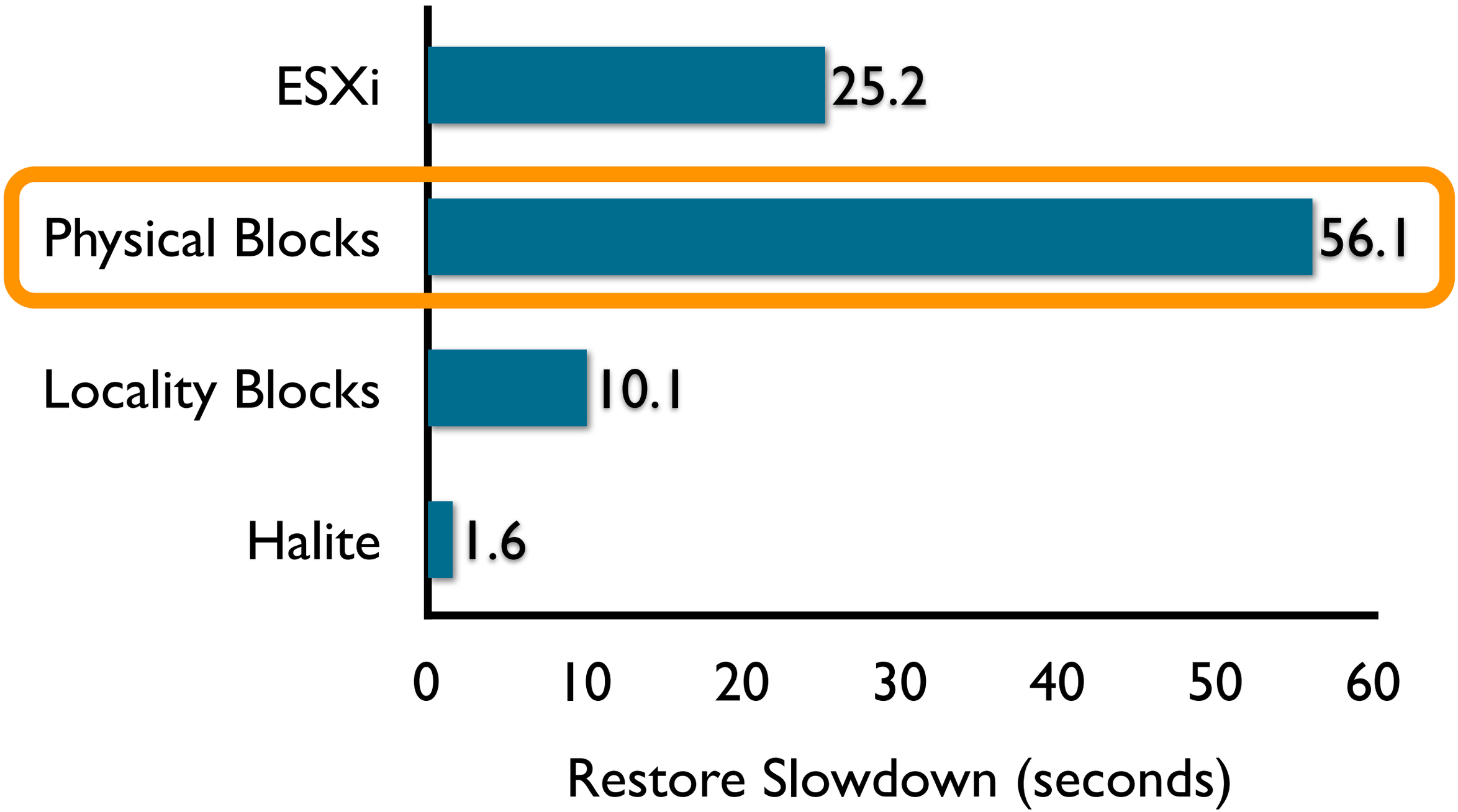
Locality blocks provide more than half of Halite's performance improvement.



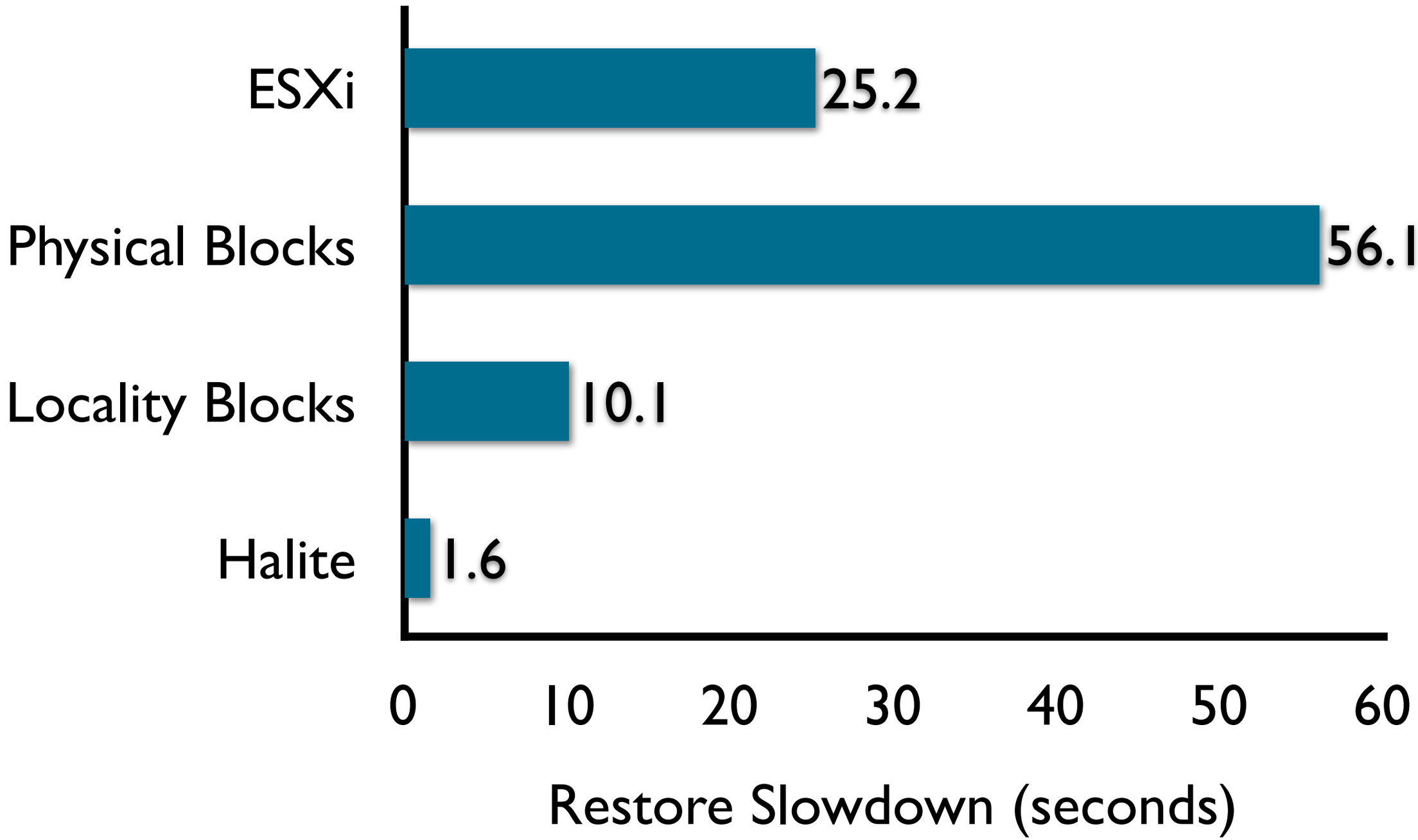
Blocks with no locality hurt performance.



Blocks with no locality hurt performance.

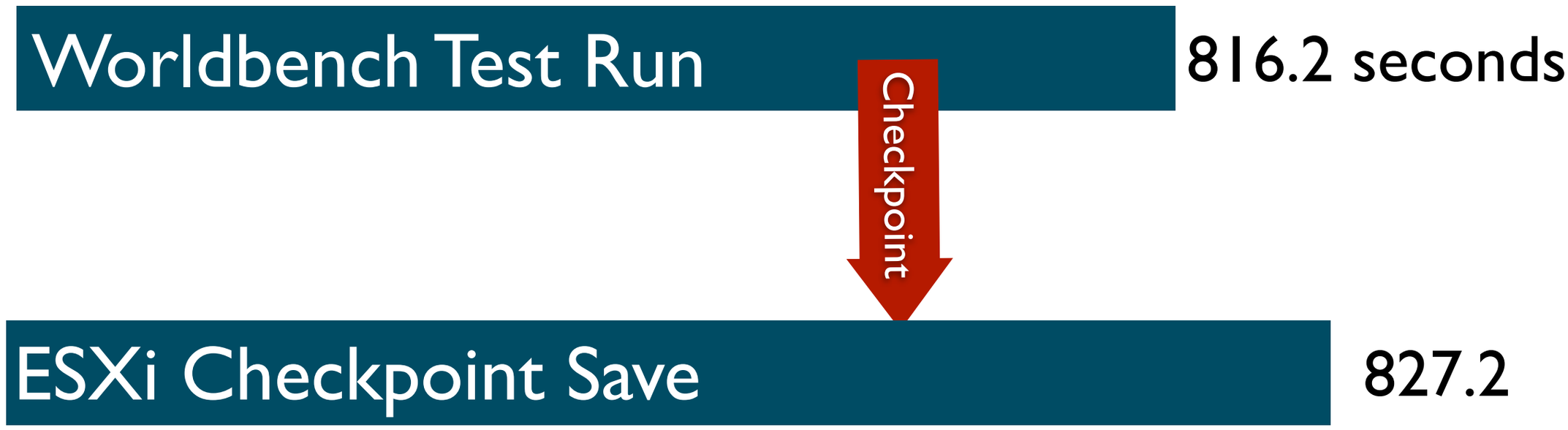


Blocks with no locality hurt performance.

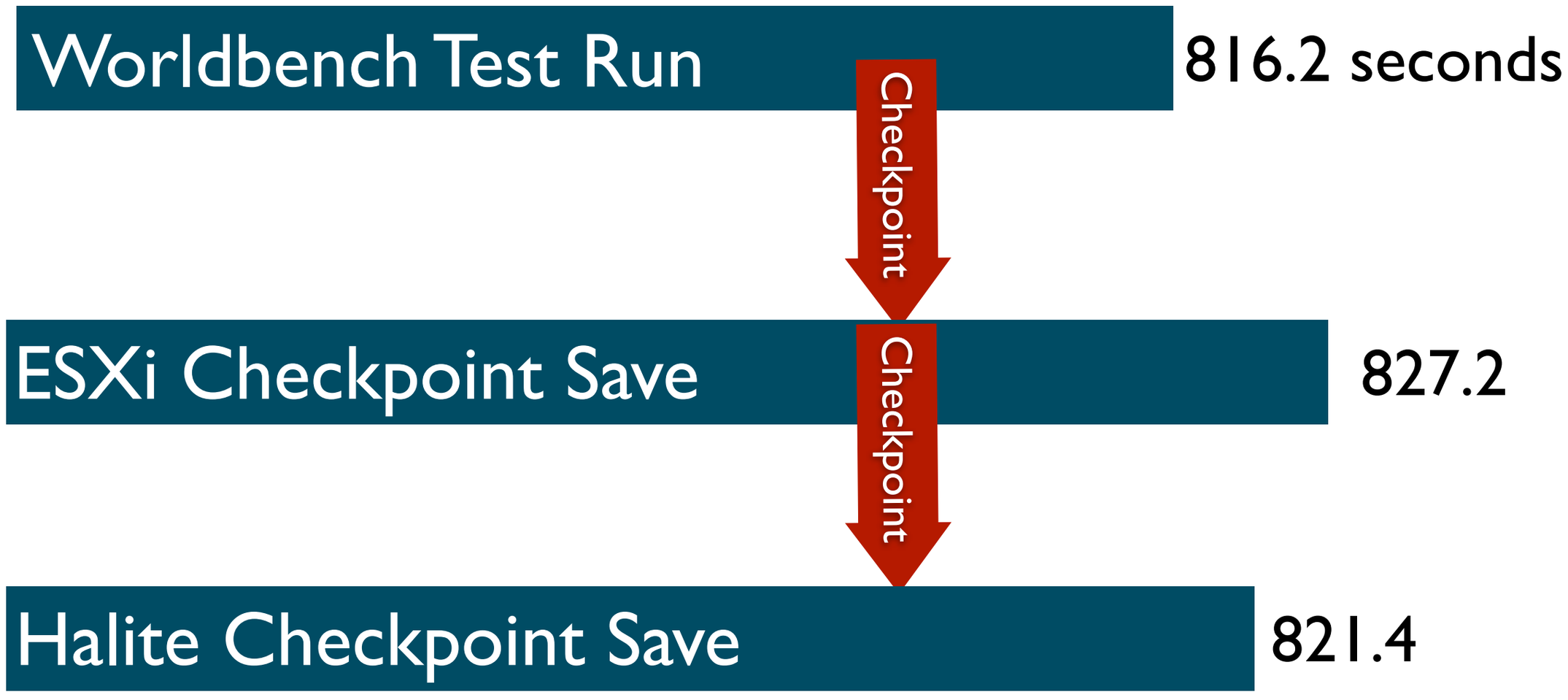


Worldbench Checkpoint Test Setup

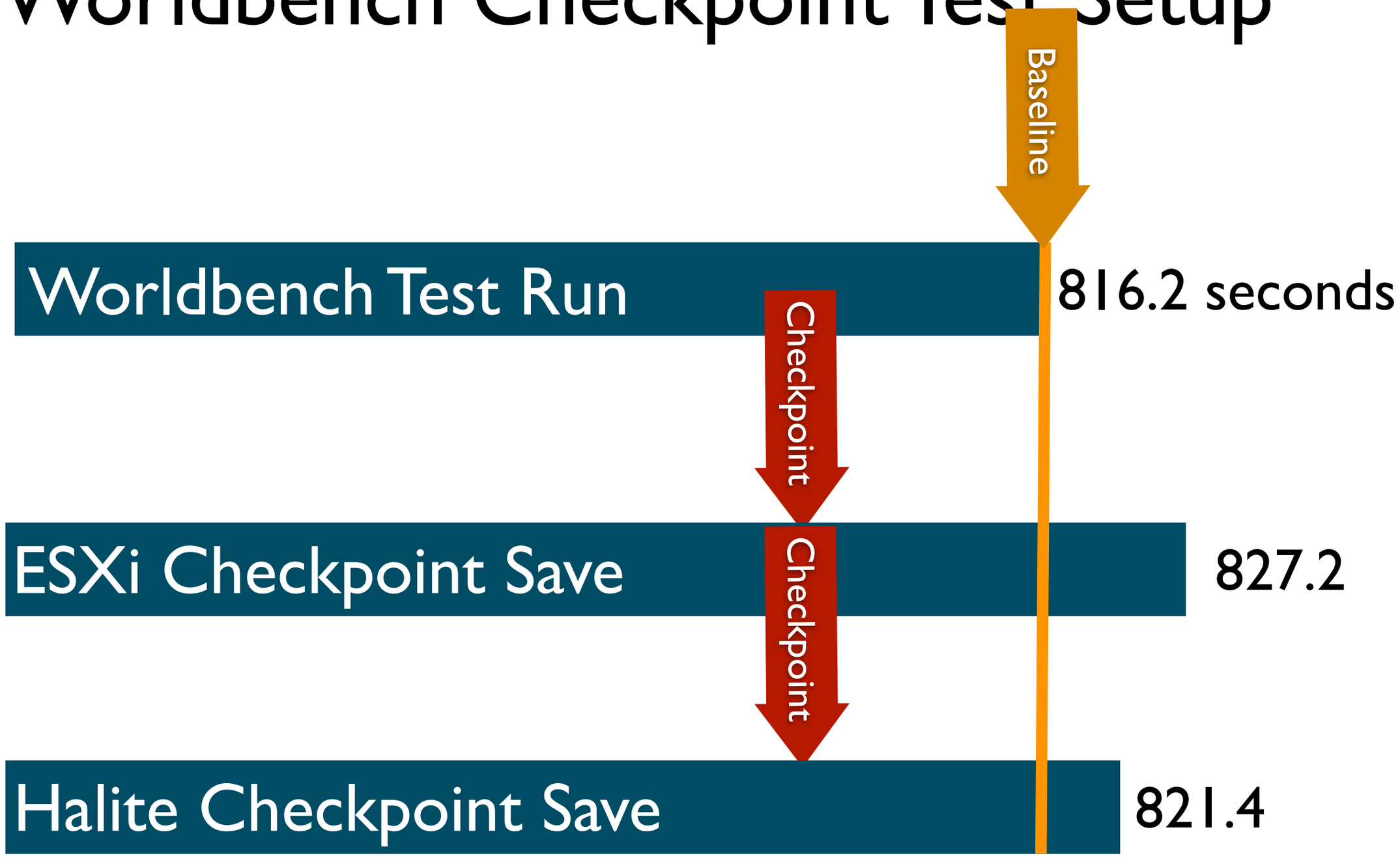
Worldbench Checkpoint Test Setup



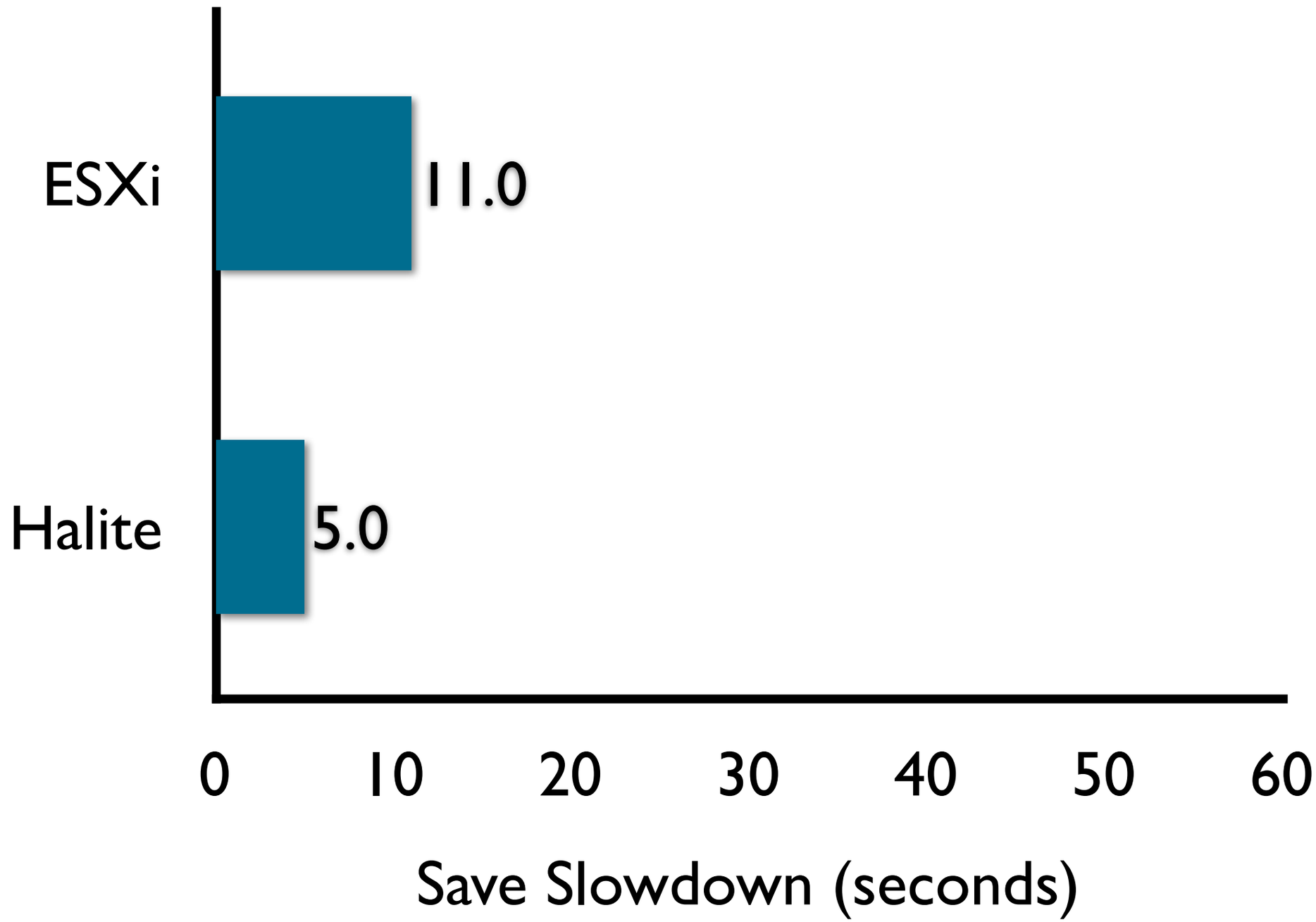
Worldbench Checkpoint Test Setup



Worldbench Checkpoint Test Setup



For Worldbench, Halite improves checkpoint save performance.



Halite

- Fast checkpoint restore is important for a new class of applications.
- Predicting access locality is better than predicting accesses for complex VM workloads.
- Halite can restore a checkpointed Windows VM in 1.6 seconds.