



Live VM Migration

PRESENTED BY: **PEYMAN NAVIDI**

Outline



- ▶ **Introduction**
- ▶ **Live VM migration approaches**
 - ▶ Pre-copy
 - ▶ Pre-copy VM with persistent states (disk) in Wide-Area Network
 - ▶ Post-copy
- ▶ **Conclusion**

Papers

- ▶ **Live Migration of Virtual Machines**

Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hanseny, Eric July, Christian Limpach, Ian Pratt, Andrew Warfield. 2004

- ▶ **LiveWide-Area Migration of Virtual Machines Including Local Persistent State**

Robert Bradford, Evangelos Kotsovinos, Anja Feldmann, Harald Schi"Oberg. 2007

- ▶ **Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning**

Michael R. Hines and Kartik Gopalan. 2009

Introduce to VM Migration

▶ **VM Migration**

- ▶ Migrating an entire OS and all of its applications as one unit between physical machine.

▶ **Off-line VM Migration**

▶ Procedure

1. Pause the VM hosting machine
2. Transfer the all states of VM
3. Resume the VM in the new machine

- ▶ Main advantage: simple

- ▶ Main disadvantage: long downtime

▶ **Live VM Migration**

- ▶ Transfer the state of VM with minimum service disruption
- ▶ Key advantage: user-invisible downtime with fast network

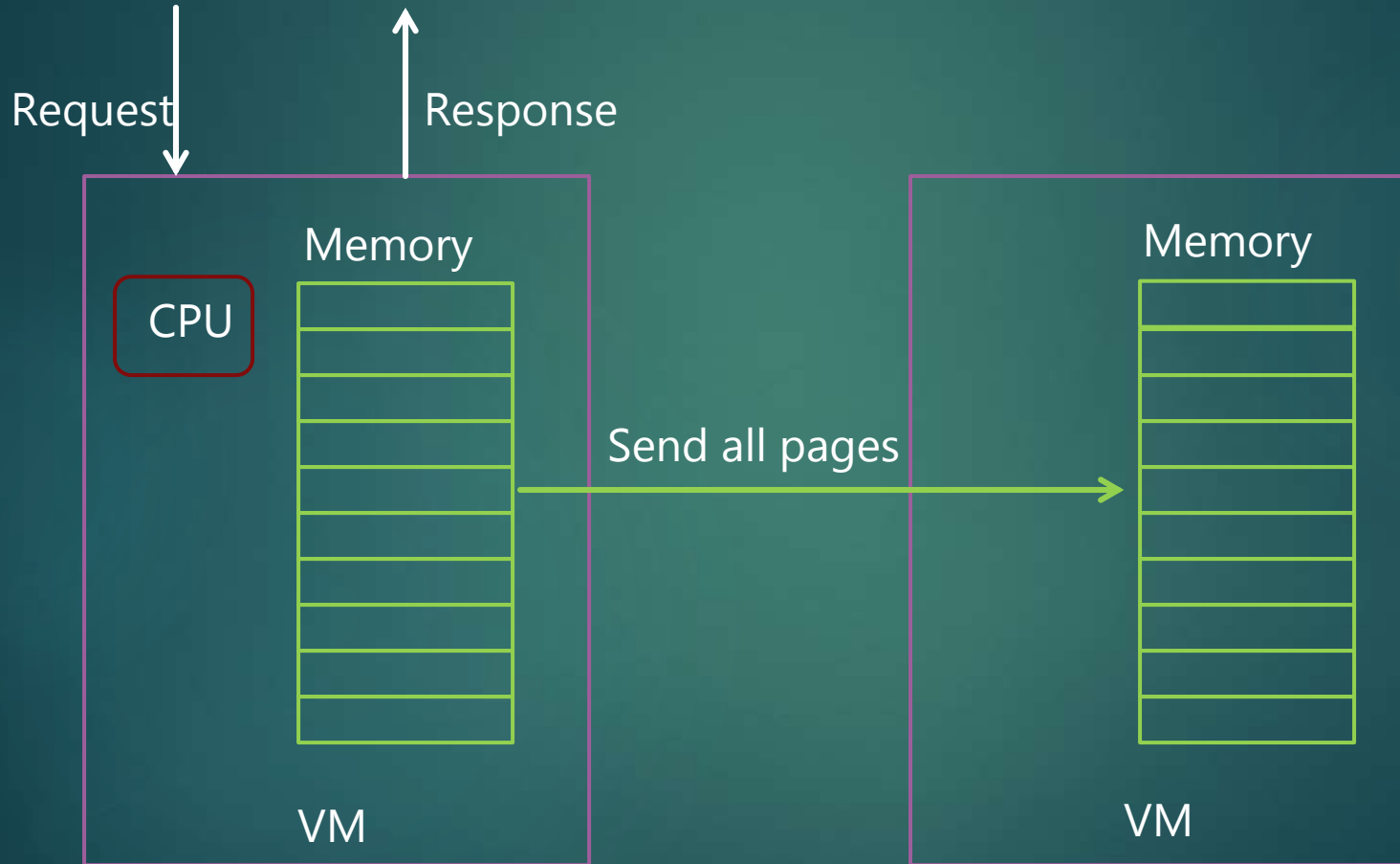
Performance metric

- ▶ **Downtime:** The time during which the migrating VM's execution is stopped.
- ▶ **Pages transferred:** The total count of memory pages transferred.
- ▶ **Total migration time:** The total time during migration.
- ▶ **Application degradation:** The extent to which migration slows down the applications executing within the migrating VM.

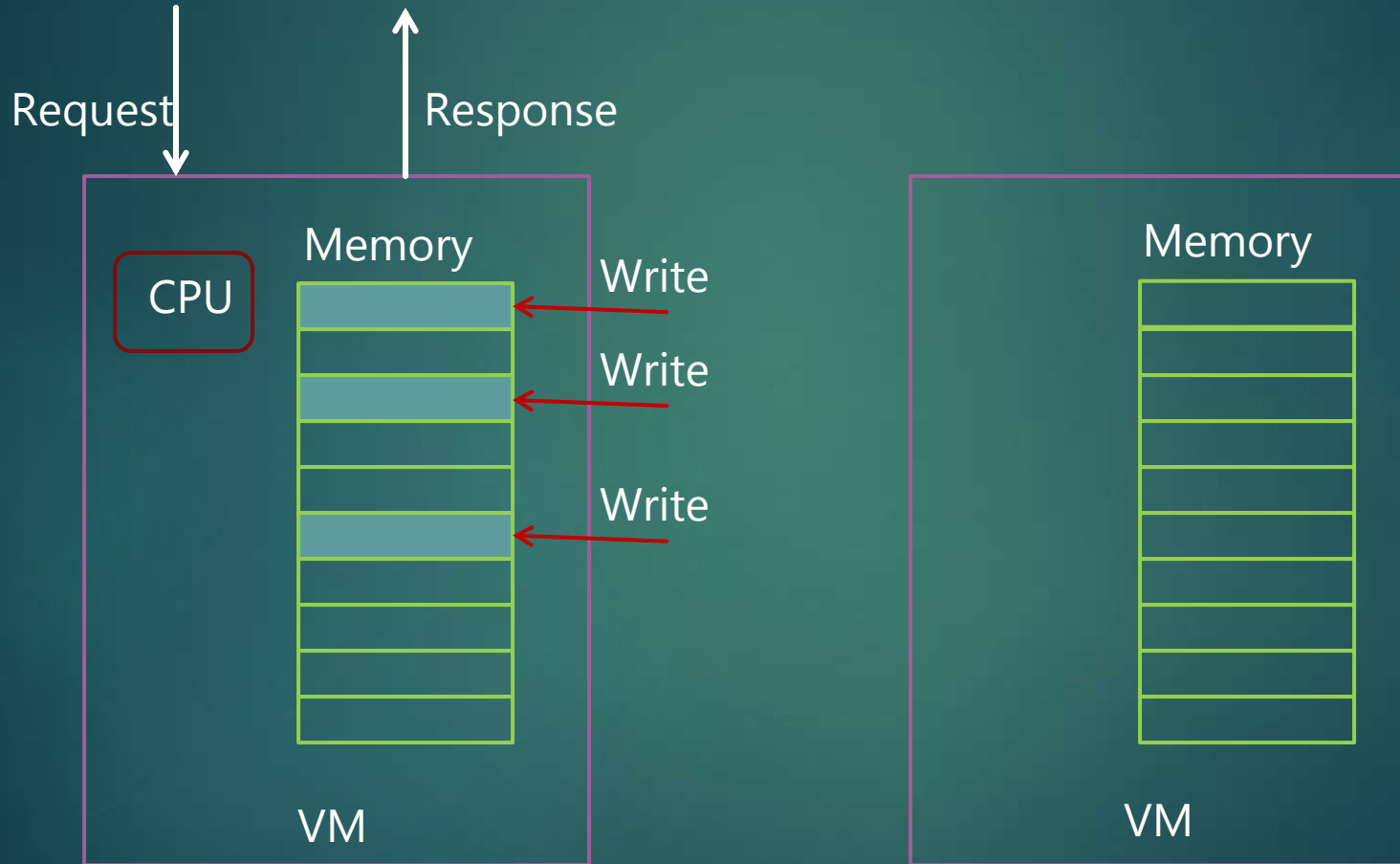
Pre-copy approach

- ▶ Assume in datacenter environment, which means a distributed file system is available, so we don't need to migrate the persistent state of VM.
- ▶ The bottleneck is memory migration (due to large in size)
- ▶ The first paper proposed a *pre-copy* approach
 1. Send all pages of memory to the targeted machine (without stopping the VM)
 2. Keep track of the pages which is updated after the last send and re-send them
 3. Repeat step 2 until the stopping criterion is met
 4. Pause the VM and transfer the rest of states (CPU states, remaining un-send pages of memory)
 5. Resume the VM in the target machine

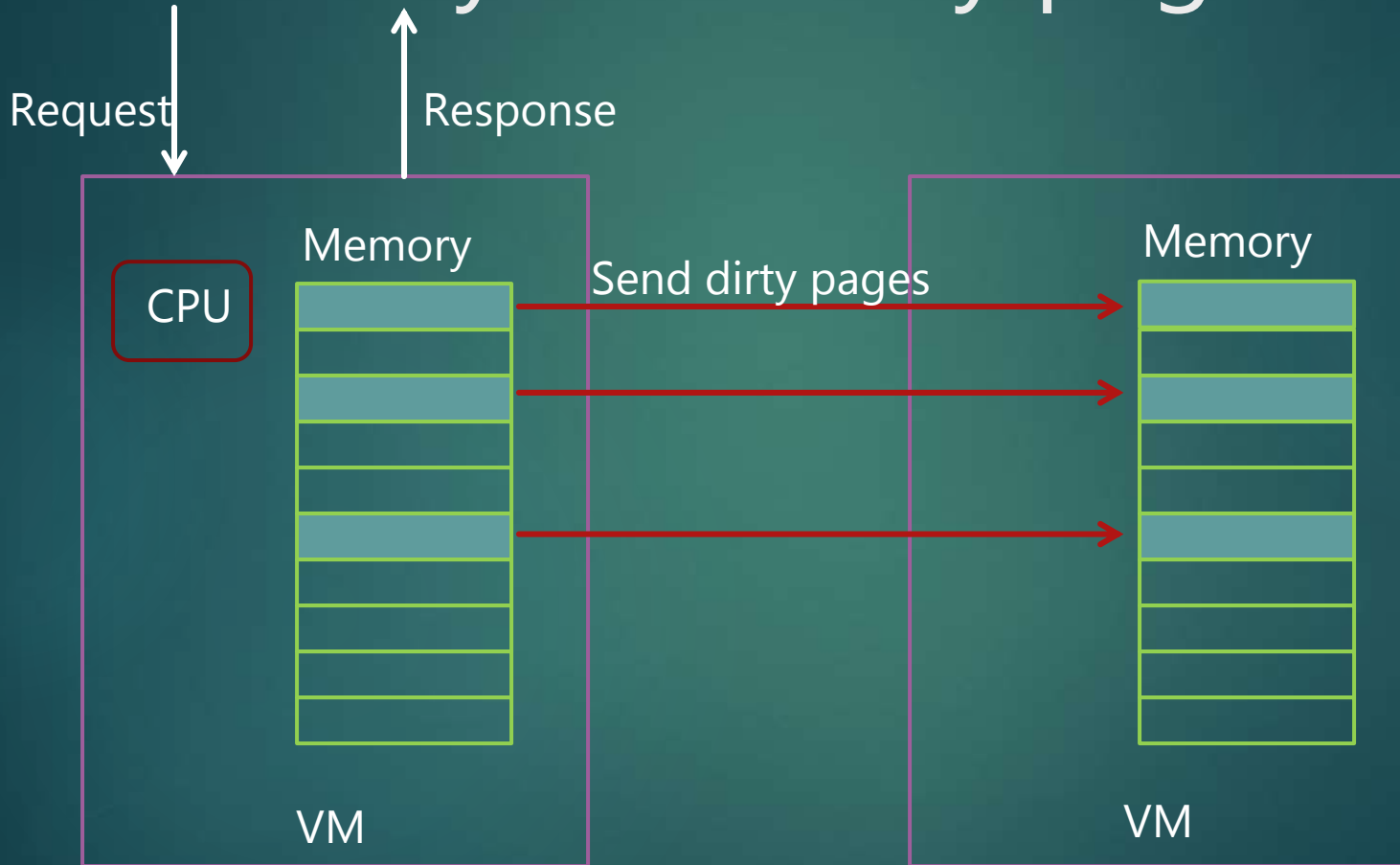
First Round: Send all pages in memory



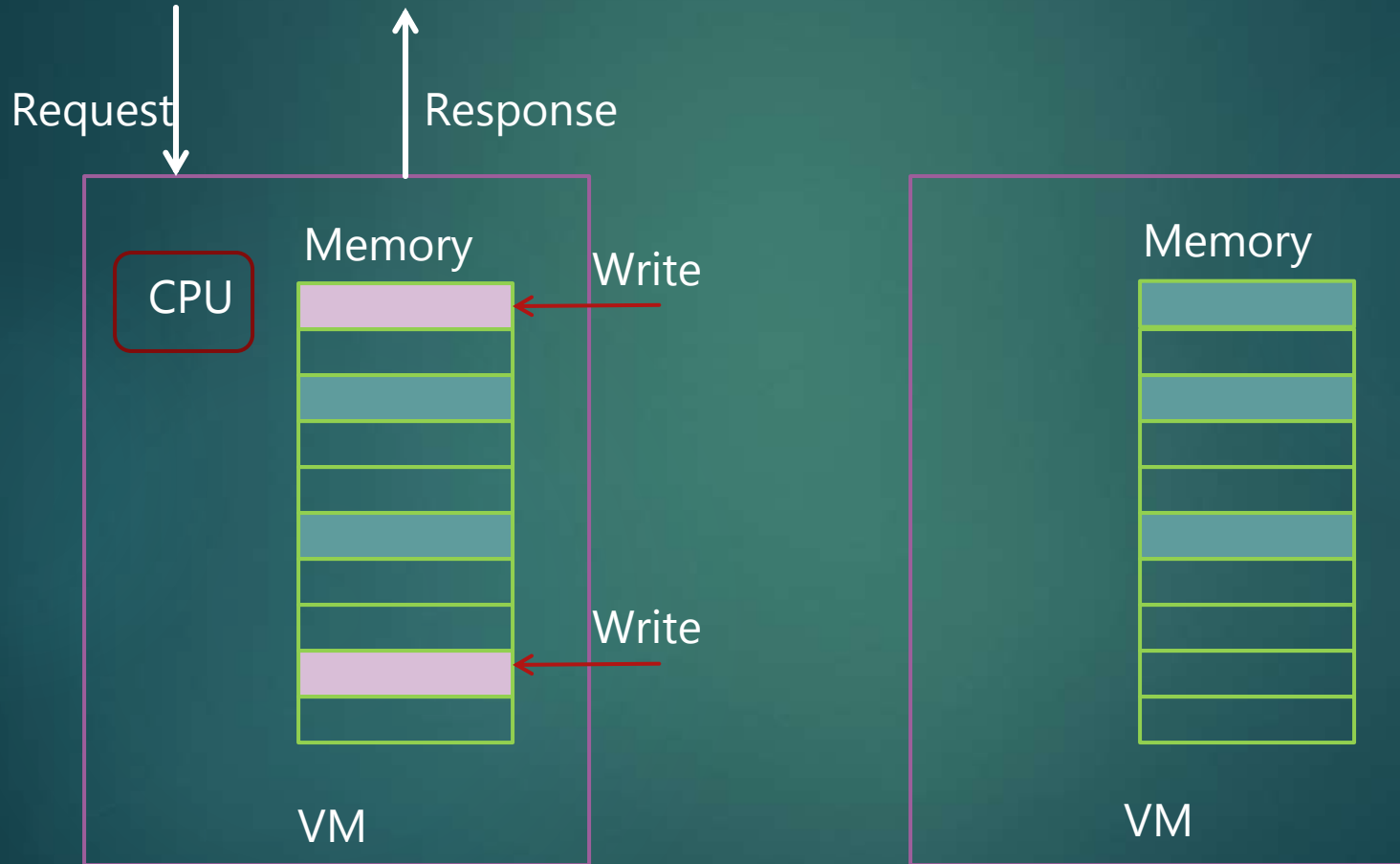
Write after Last-send



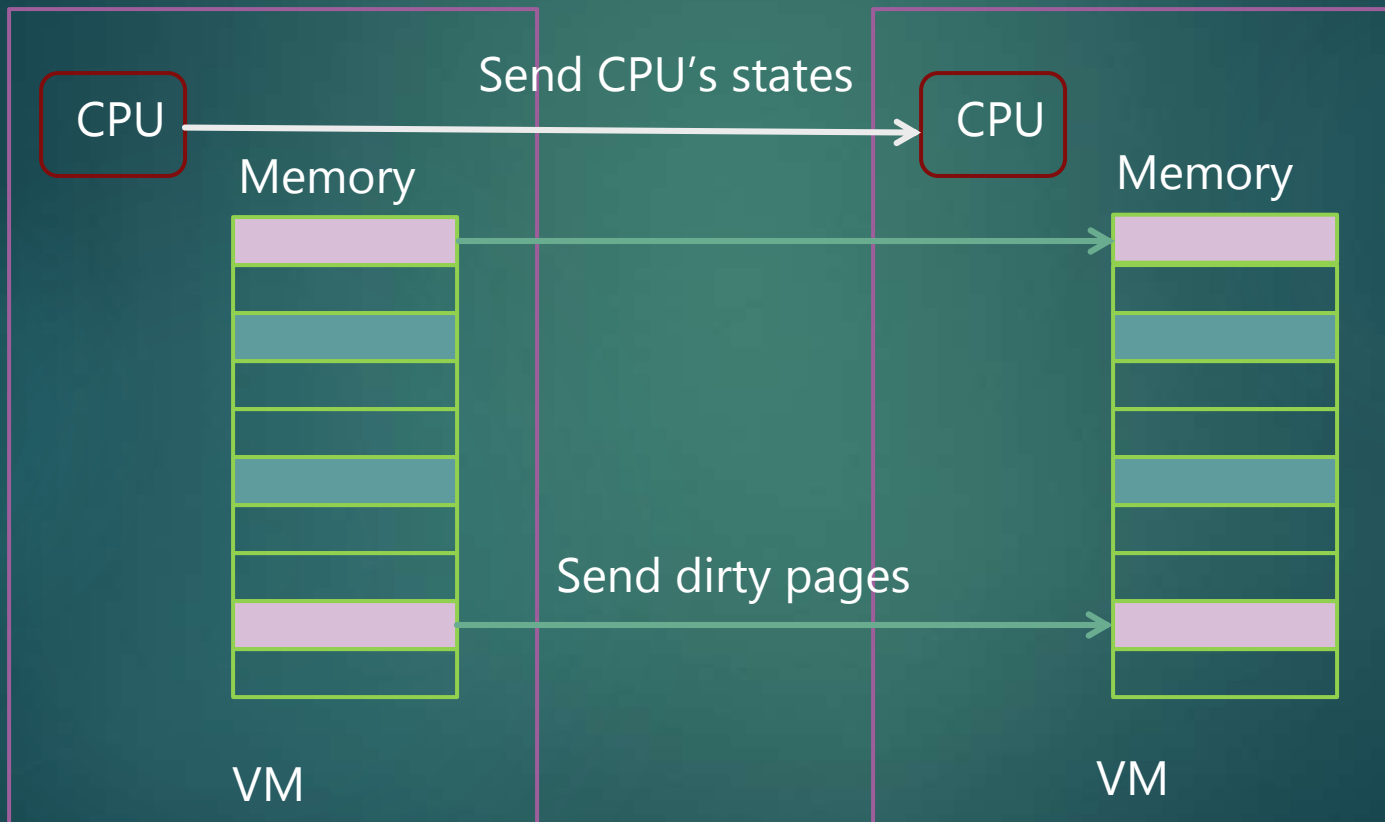
Subsequent Rounds: Iteratively send dirty pages



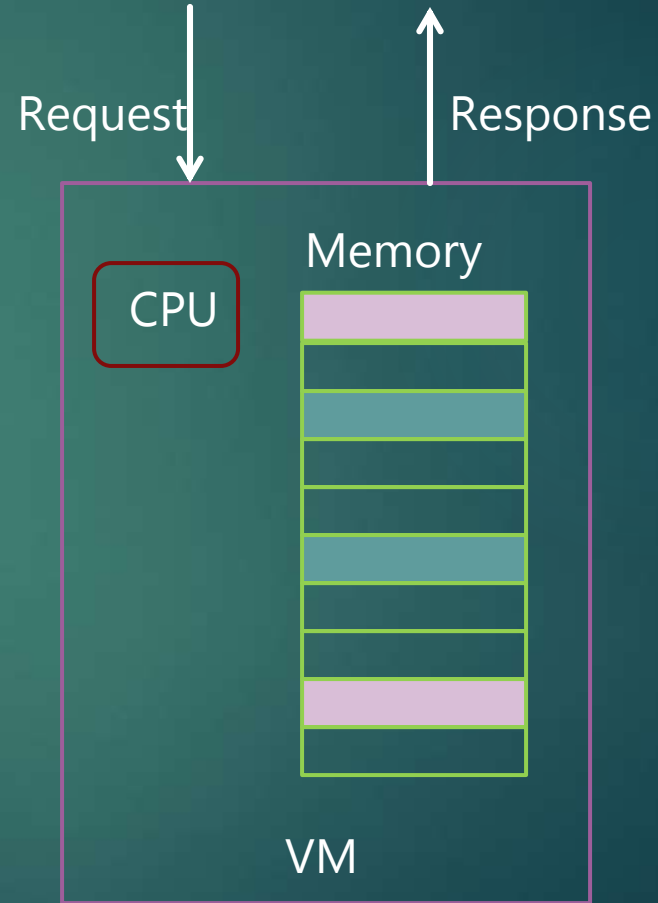
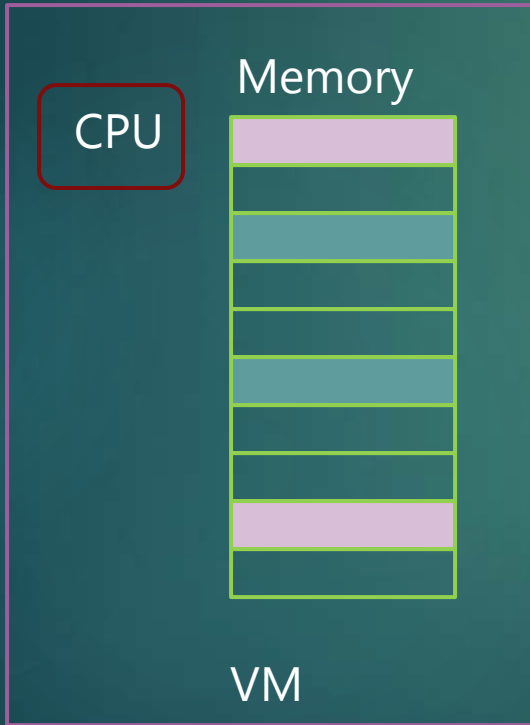
Write after Last-send



Stop the VM and send



Resume the VM



Pre-copy Evaluation

▶ Evaluation result

- ▶ Reported by this paper
 - ▶ The reported experiments were conducted in a fast network environment
 - ▶ Service downtime: about 200ms
 - ▶ Total migration time: about 70sec
- ▶ Reported by another paper
 - ▶ LAN: 3sec
 - ▶ WAN: 68sec

▶ Pros

- ▶ Minimal application degradation
- ▶ Reasonable downtime and total migration time in fast network
- ▶ VMware and Xen uses this approach

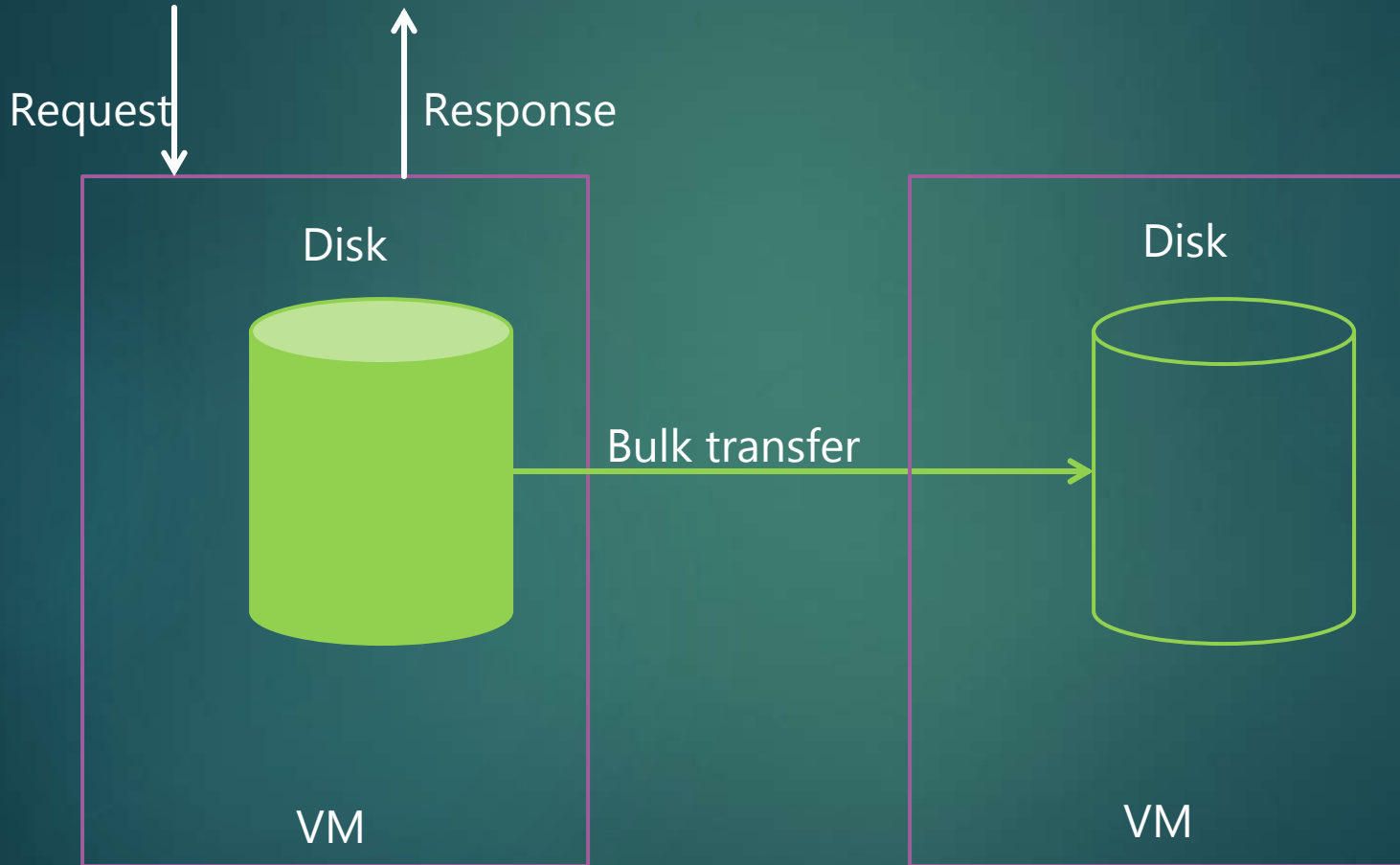
▶ Cons

- ▶ It won't work very well in write-heavy workload or slow network
- ▶ It generates higher network traffic

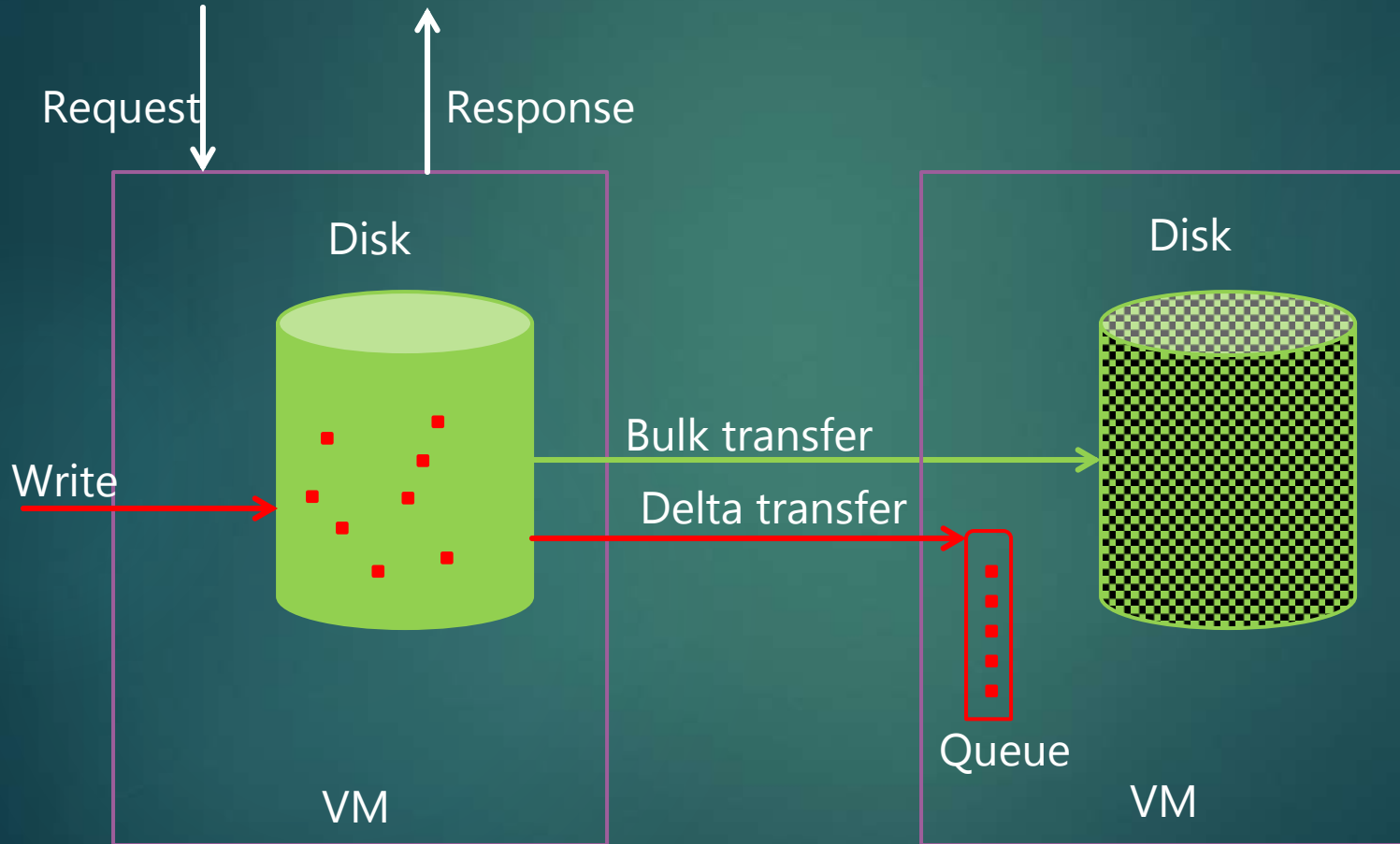
Live Wide-Area VM Migration

- ▶ The first paper discussed transferring run-time states of a VM within datacenter.
- ▶ This second paper expands on that to transfer the VM's image, its persistent state and on-going network connections over a WAN as well.
- ▶ By combining pre-copying, write-throttling and a block-driver, we can achieve this.
 - ▶ Write-throttling : **threshold** and a **delay**
 - ▶ Block-driver : XEN block backend drivers

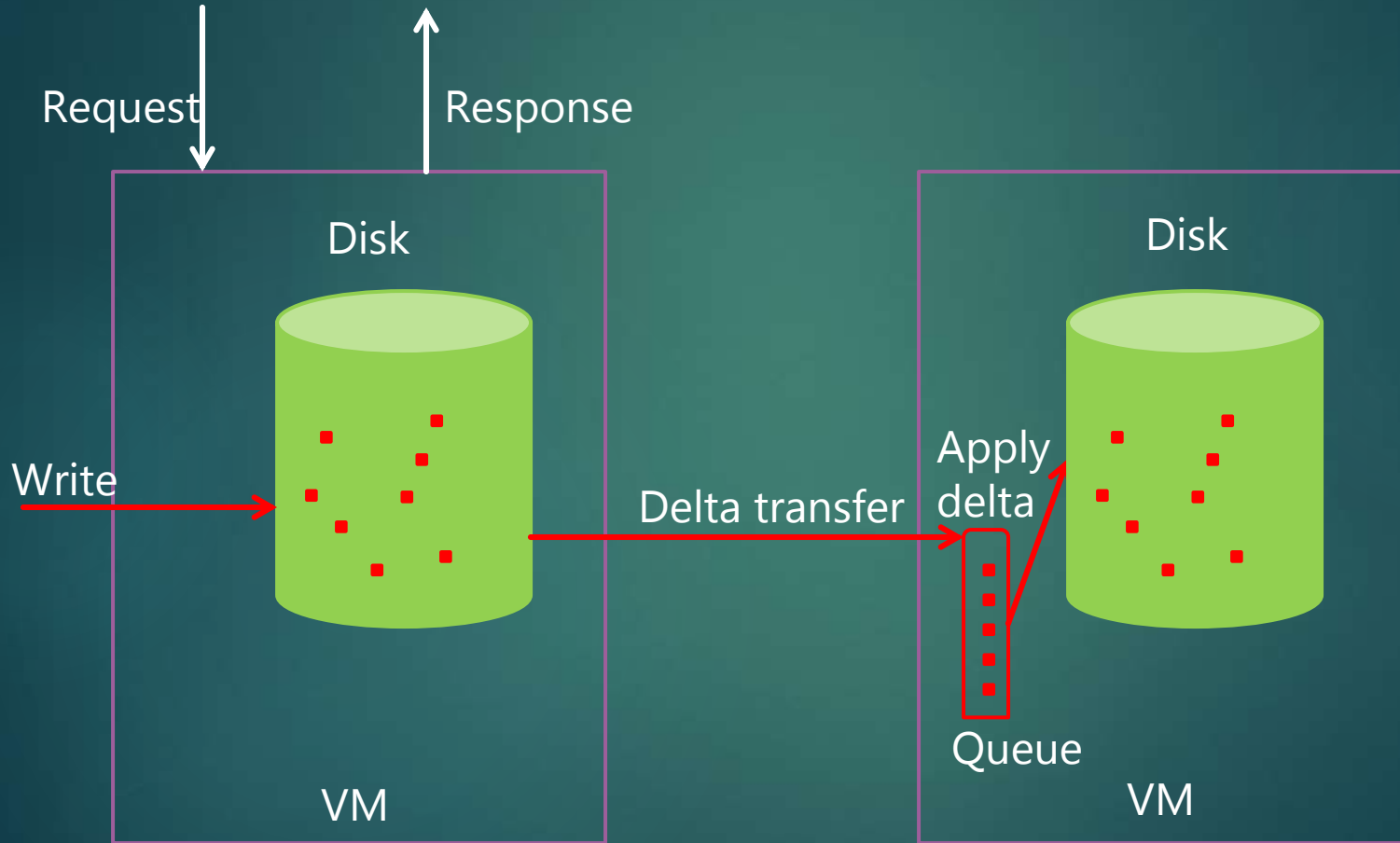
Live Disk Migration



Record Writing to Disk and Transfer



Apply Delta



Disk Pre-copy Evaluation

▶ **Experiment result**

▶ Local Network

- ▶ Service downtime: about 3sec
- ▶ Total migration time: about 200sec

▶ Wide-Area Network

- ▶ Service downtime: about 68sec
- ▶ Total migration time: about 3600sec

▶ **Pros**

- ▶ Minimum service degradation during migration

▶ **Cons**

- ▶ Long migration time for large disk
- ▶ Long migration time for write-intensive workload

Post-copy Approach

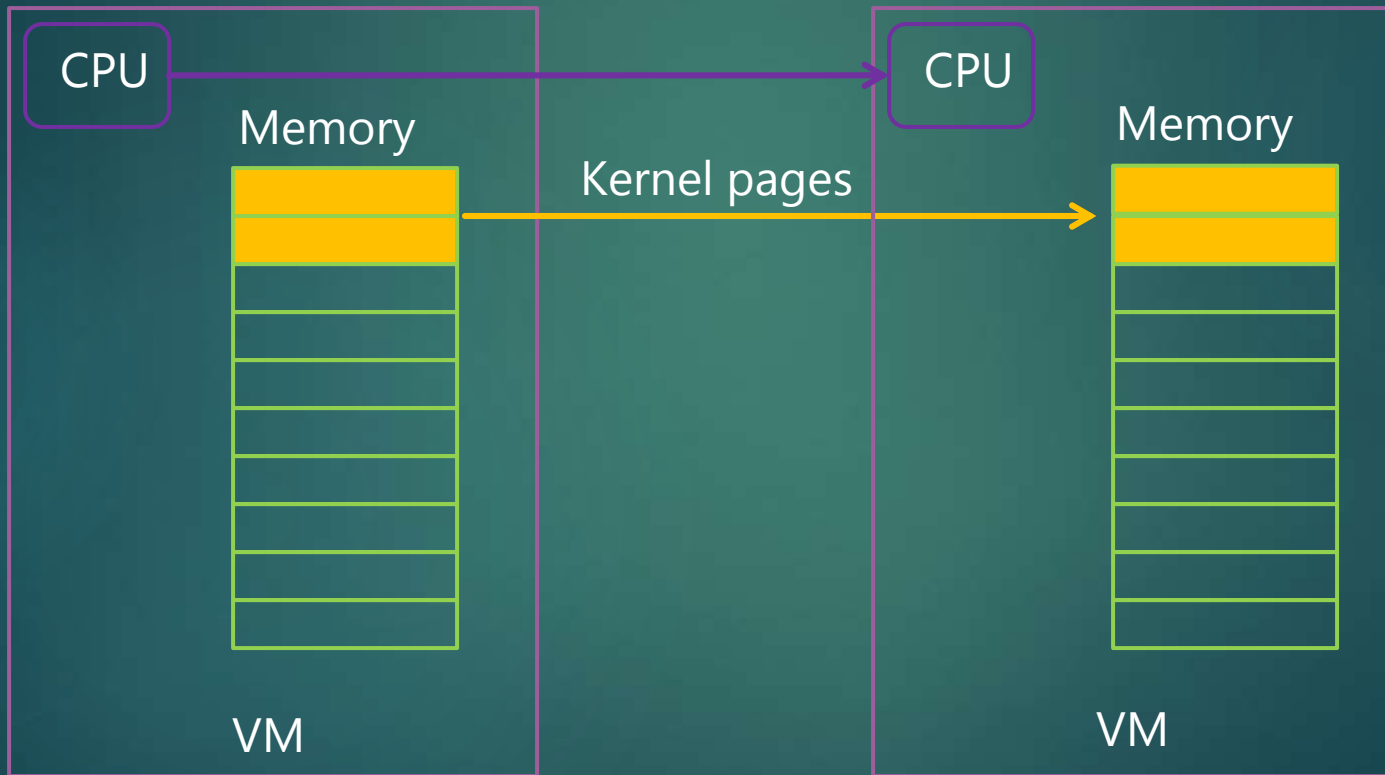
▶ **Pre-copy**

- ▶ Copy memory states first, then CPU
- ▶ Key advantage: avoid service degradation during migration

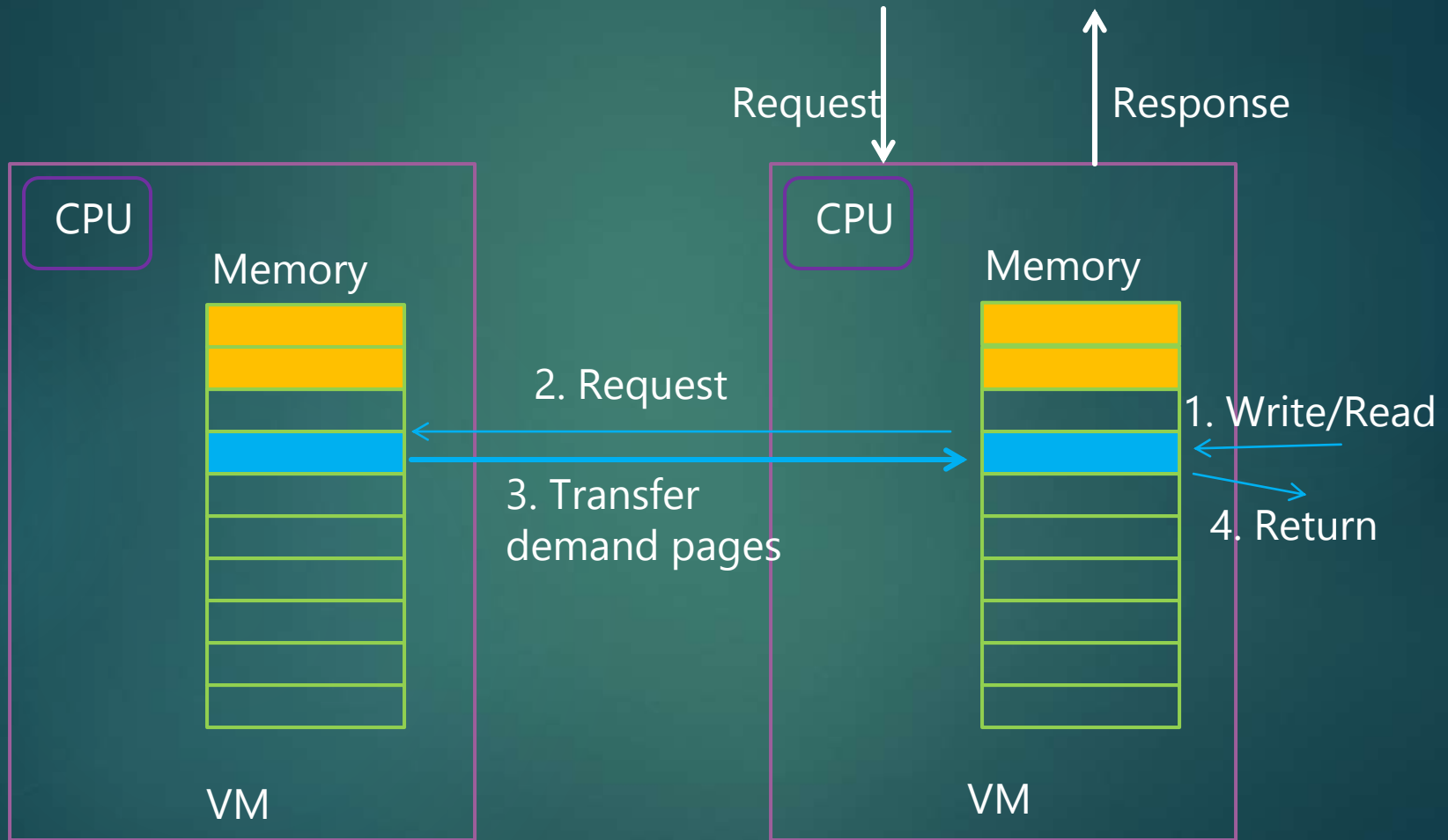
▶ **Post-copy**

- ▶ Copy CPU states first, then copy memory pages on demand (called demand paging)
- ▶ Key advantage: minimum network traffic
- ▶ Disadvantage: demand page over the network significantly reduces the performance of applications (need to fix)
 - ▶ Active pushing: keep sending undemand pages from source to target
 - ▶ Pre-paging: sending the pages near the demand page as well
 - ▶ Self-ballooning: don't send unallocated pages

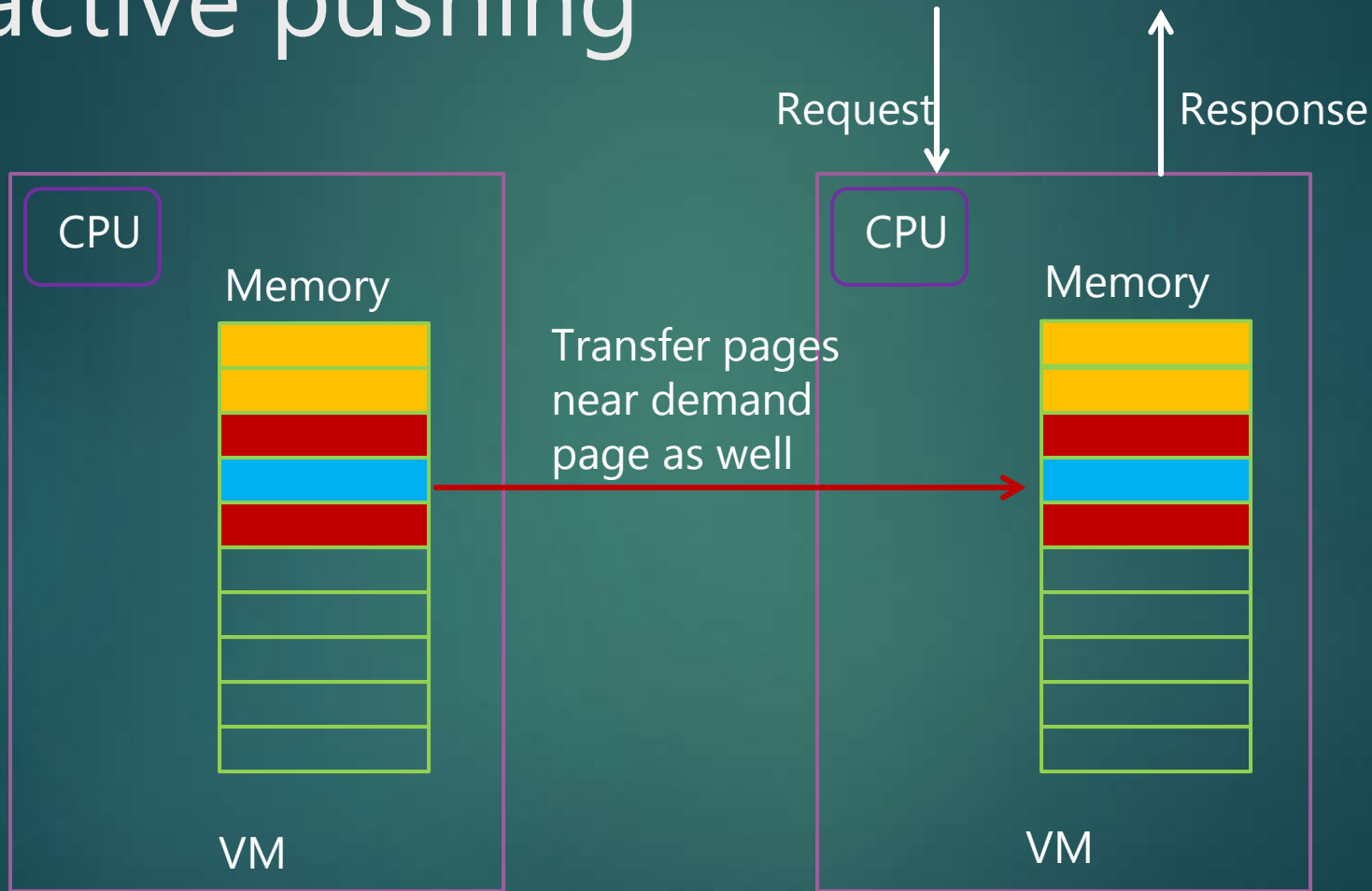
Post-copy: stop VM and transfer CPU and kernel pages



Post-copy: demand paging



Post-copy: pre-paging & active pushing



Post-copy Evaluation

▶ **Evaluation result**

- ▶ Service downtime: about 300% larger than pre-copy (due to known issue of their implementation)
- ▶ Total migration time: about 40% smaller than pre-copy

▶ **Pros**

- ▶ Low network traffic
- ▶ Low total migration time

▶ **Cons**

- ▶ Application performance degradation during migration

Conclusion

- ▶ Pre-copy is the mainstream live VM migration techniques
- ▶ The downtime due to VM migration using pre-copy is sensitive to the speed of network
- ▶ Current live VM migration techniques didn't work very well in Wide-Area network
- ▶ Post-copy can be used to reduce the network traffic during migration but degrade the performance of application