Virtual Machine Migration

Pierre Riteau

University of Rennes 1, IRISA
Inria Rennes - Bretagne Atlantique

October 21, 2011
Journée Cloud France Grilles, Lyon

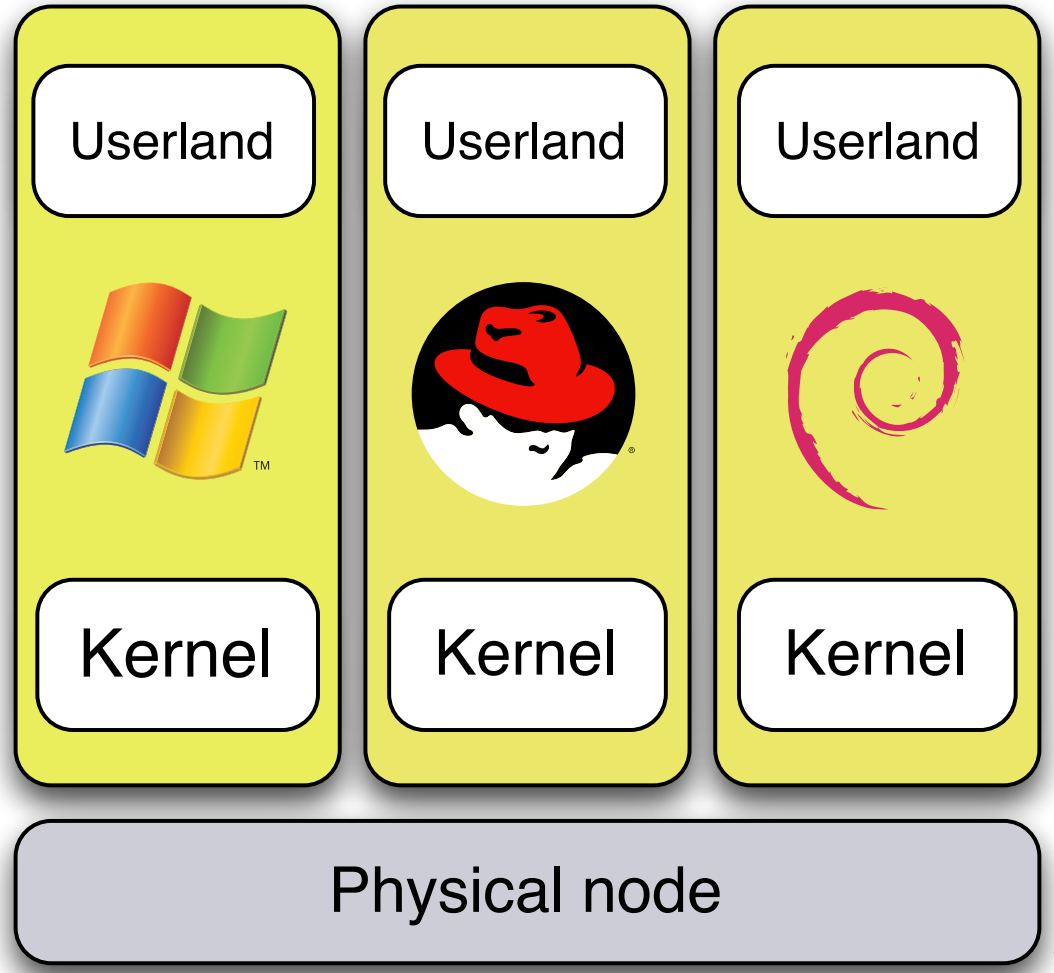
Outline

Outline

Hardware Virtualization

- Presents a computer similar to a real physical one
- With CPU(s), memory, disk(s), network interface(s), etc.
- The virtual machine runs a full OS
- Full Virtualization vs Paravirtualization
- Hypervisors: VMware, Xen, KVM, etc.
- Virtual machines provide complete encapsulation of
  - Applications
  - Libraries
  - Operating system

Hardware Virtualization



Outline

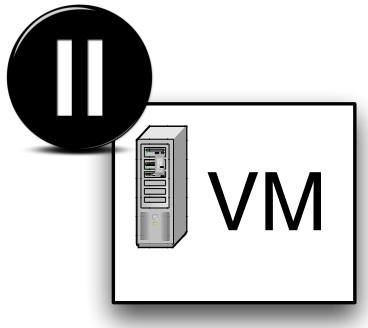
Migration

- Relocate VM from one physical host to another
- Complete encapsulation $\rightarrow$ no OS support needed
- Transfer VM state over the network
  - Processor state (CPU registers)
  - Hardware devices state (hardware registers)
  - Memory content
  - (Possibly disk content)

Pure stop-and-copy

- Simplest approach
- Suspend source VM on source host
- Copy all VM state over the network
- Resume source VM on destination host
- Used by the Internet Suspend/Resume project

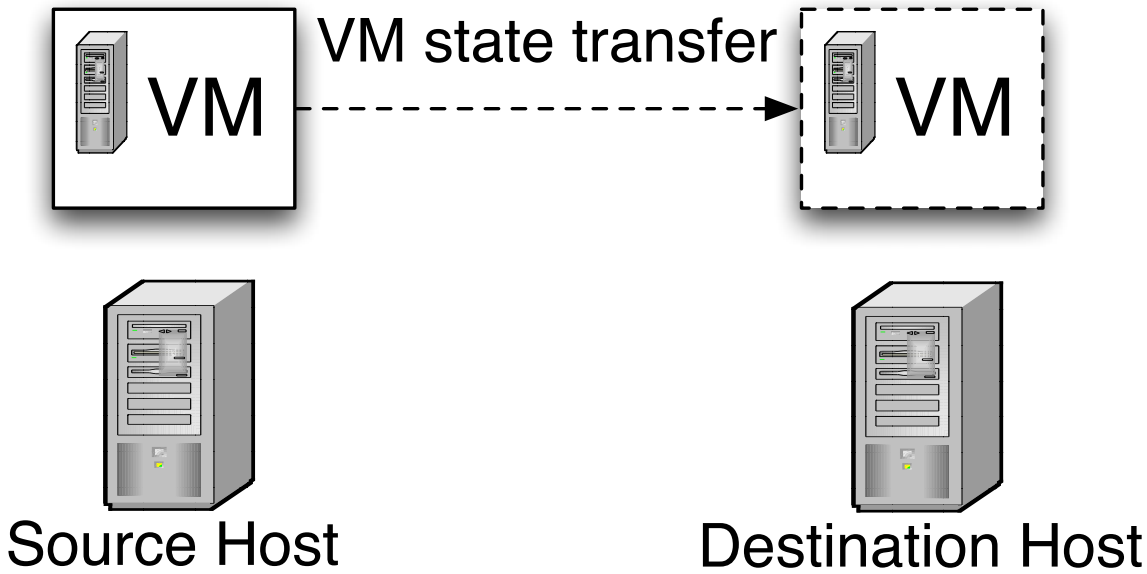
Pure stop-and-copy



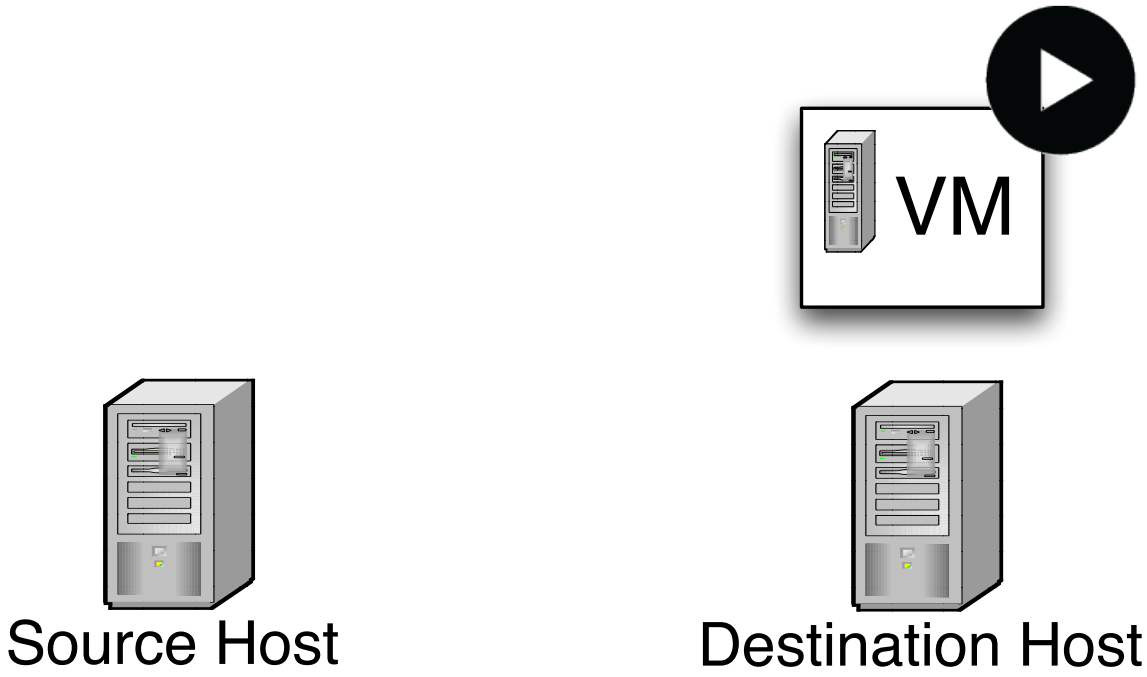
Source Host

Destination Host

Pure stop-and-copy

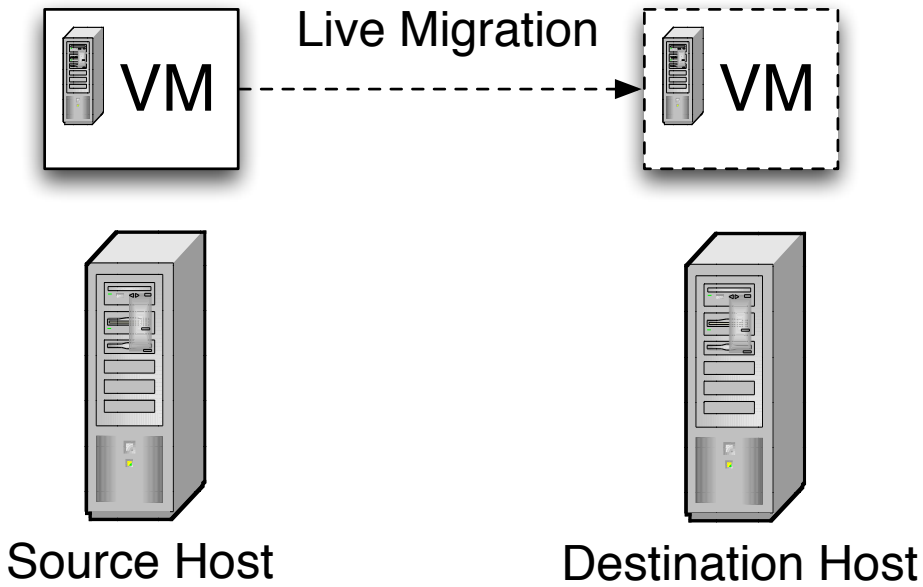


Pure stop-and-copy



Live Migration of Virtual Machines

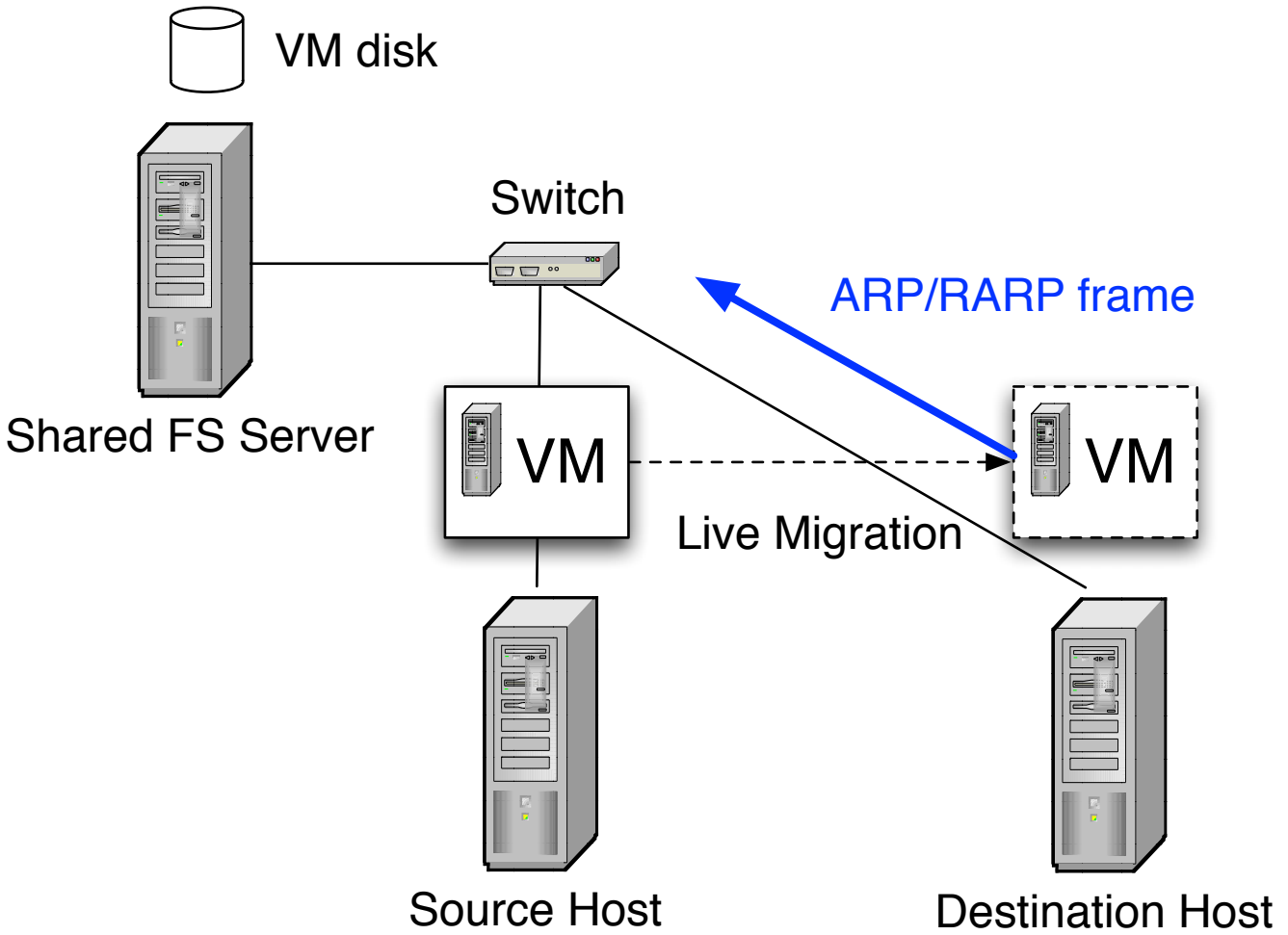
- Problem with pure stop-and-copy: long downtime
- Live migration
  - Minimize downtime (milliseconds)
  - Works by transferring state during execution



Live Migration of VMs in LANs

- Live migration initially proposed for LANs
- Clark et al., NSDI '05 & Nelson et al., USENIX '05
- Transfer from source host to destination host of the same LAN
- What about storage and network resources?
- Shared storage (e.g. NFS) $\rightarrow$ no migration needed
- Network traffic redirected with gratuitous ARP/RARP frames

Live Migration of Virtual Machines



Use Cases

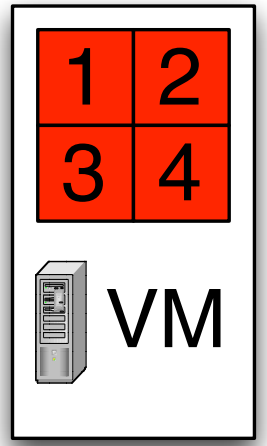
- Offers many advantages
- Load balancing / Reduced energy consumption
  - Migrate VMs in case of hotspots (c.f. Adrien Lèbre's talk)
  - Consolidate VMs on a subset of nodes
  - Turn off unused physical nodes
  - Entropy (Ecole des Mines de Nantes)
- Transparent infrastructure maintenance
- Pro-active fault tolerance
  - Detect future faults from hardware events
  - Preemptively migrate on another node
  - Nagarajan et al., SC 07

Outline

Pre-Copy Live Migration

- Traditional method used for migration of processes
- Iterative process
  - Copy all memory content to the destination host (while the VM continues running)
  - Do multiples iterations to copy modified memory pages during the previous period
  - When *enough* iterations have been done, stop the VM and
    - Copy the remaining modified memory pages
    - Copy the CPU and device state
  - Resume VM on destination host
- Method implemented by all production hypervisors

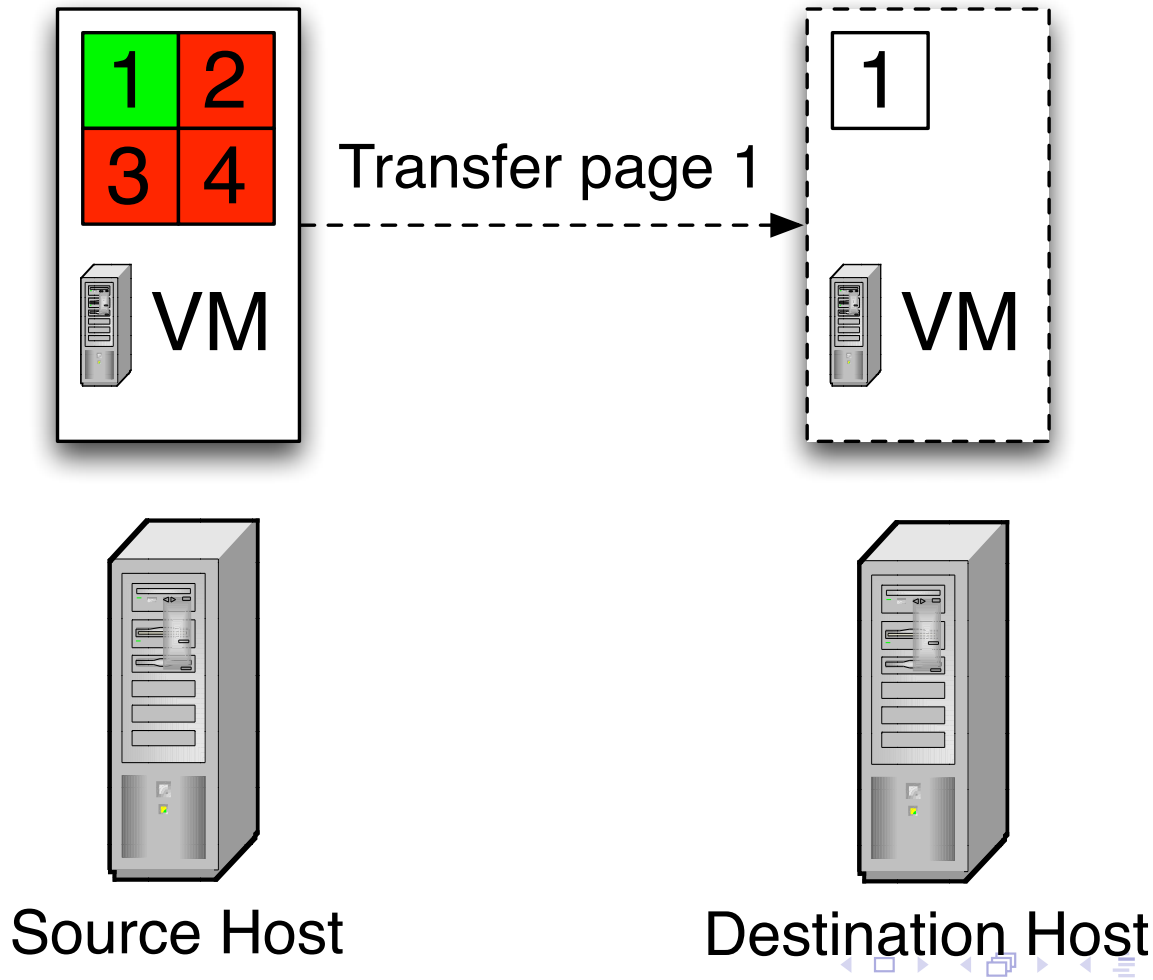
Pre-Copy Live Migration



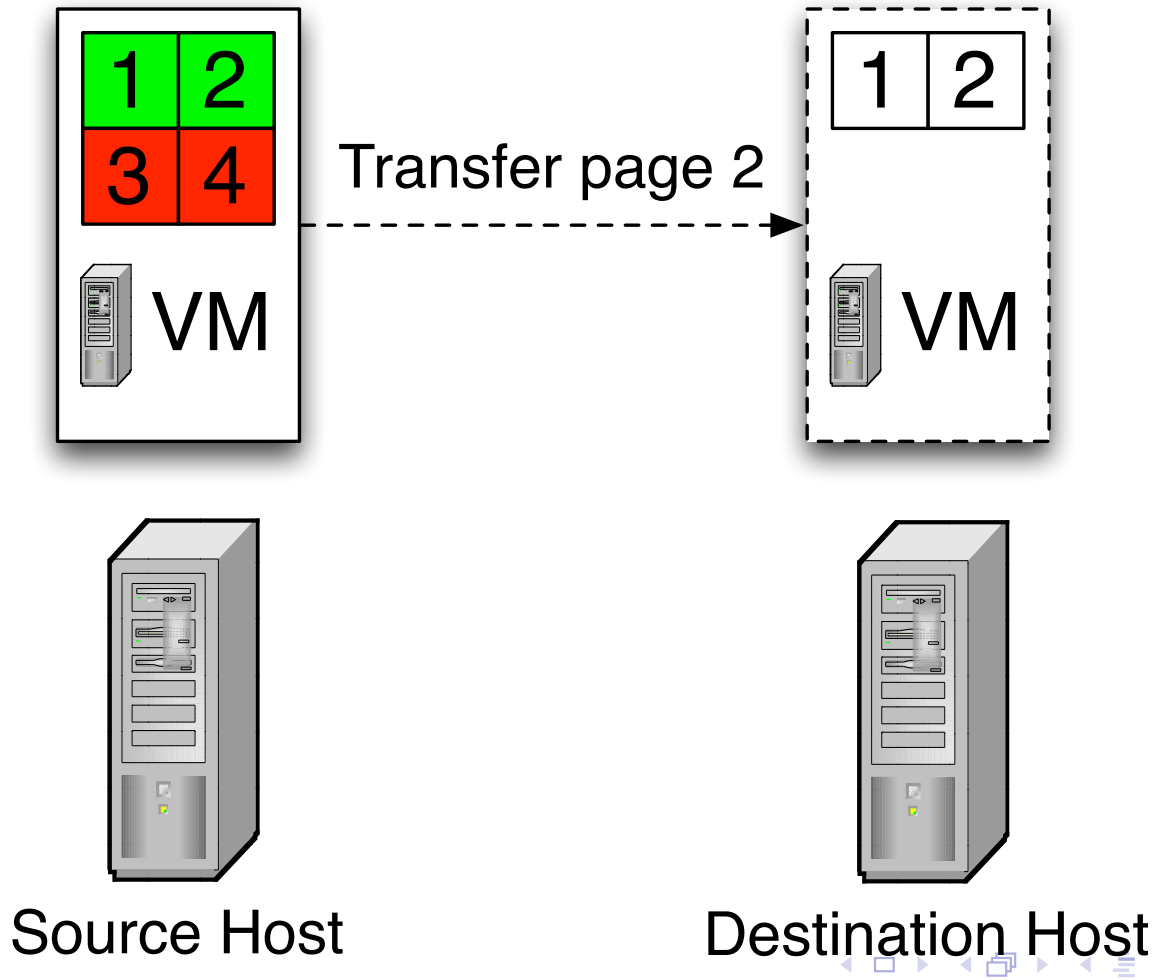
Source Host

Destination Host

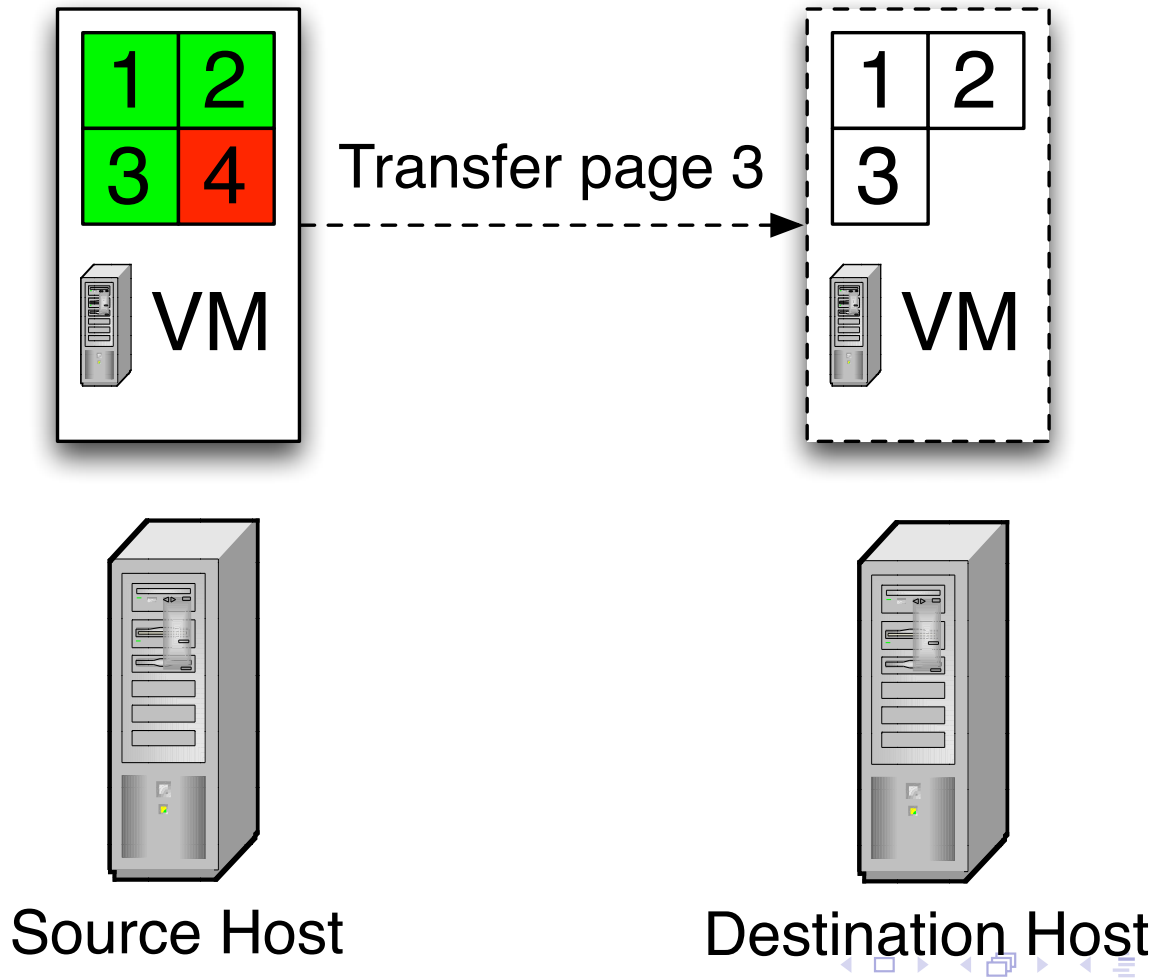
Pre-Copy Live Migration



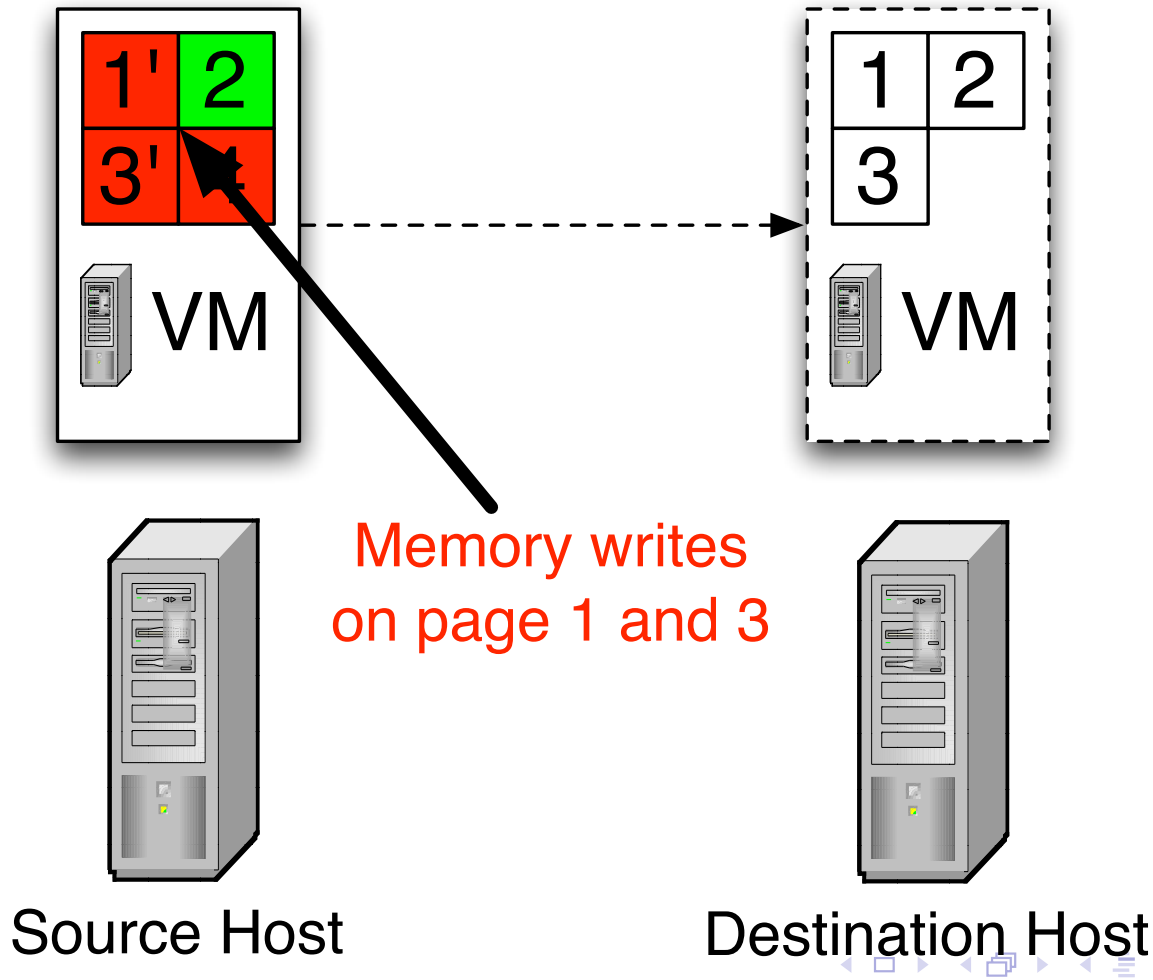
Pre-Copy Live Migration



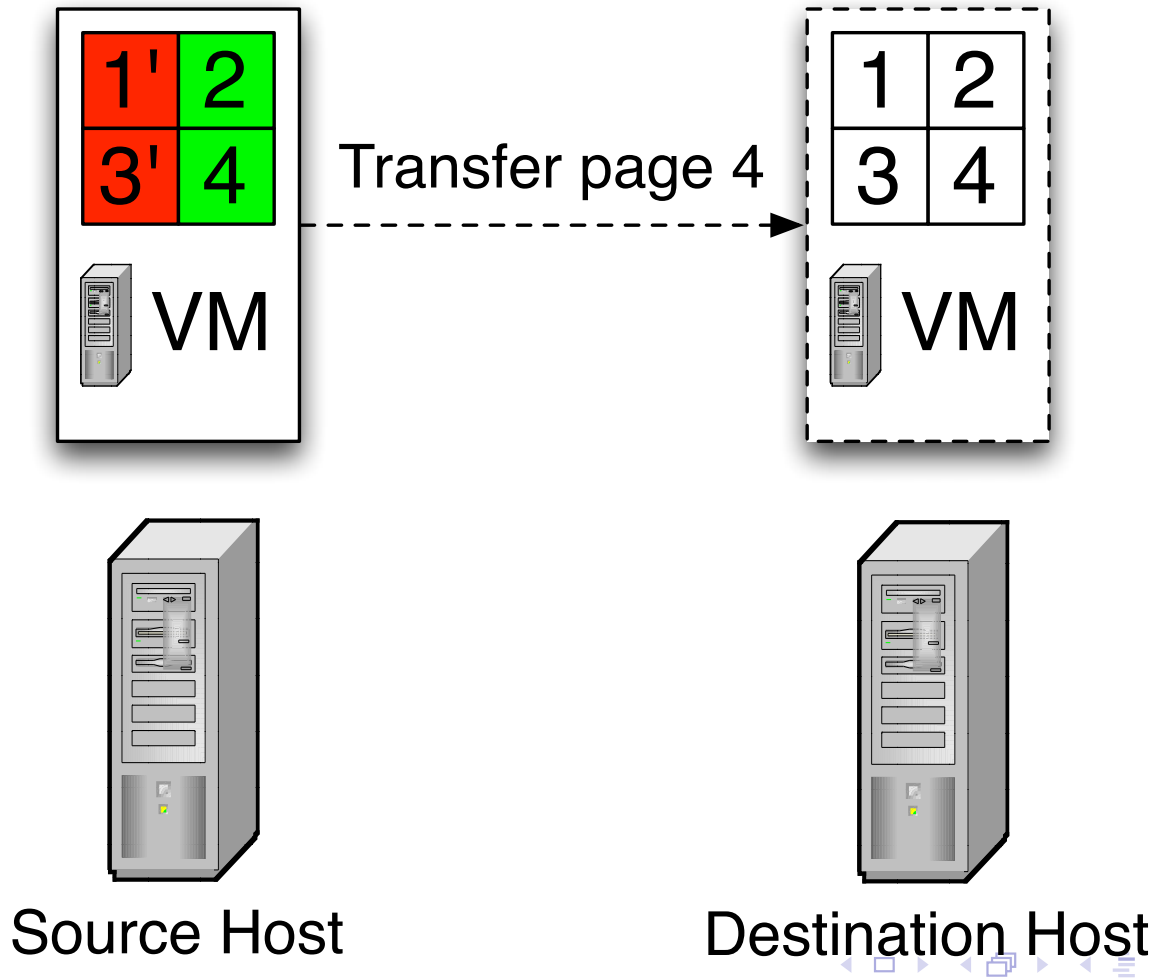
Pre-Copy Live Migration



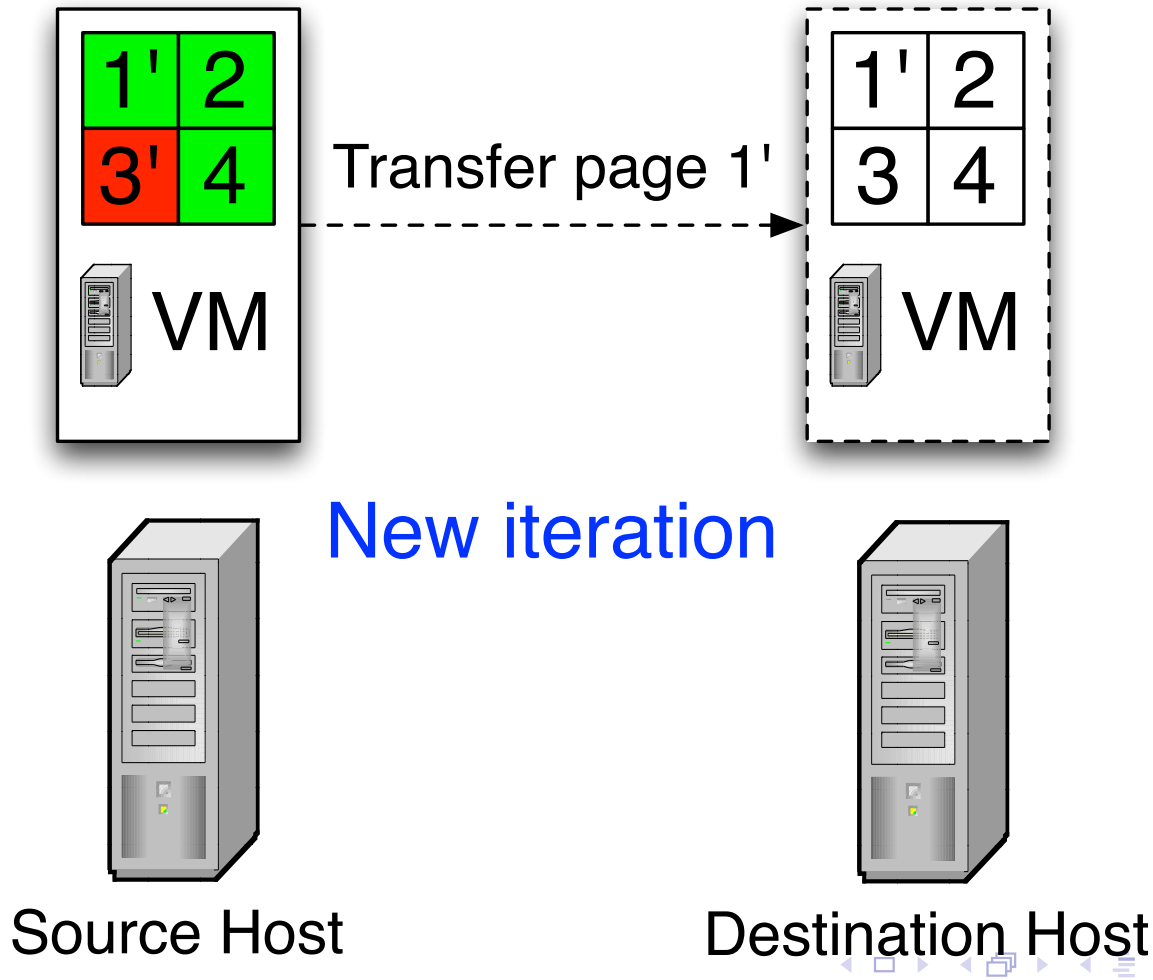
Pre-Copy Live Migration



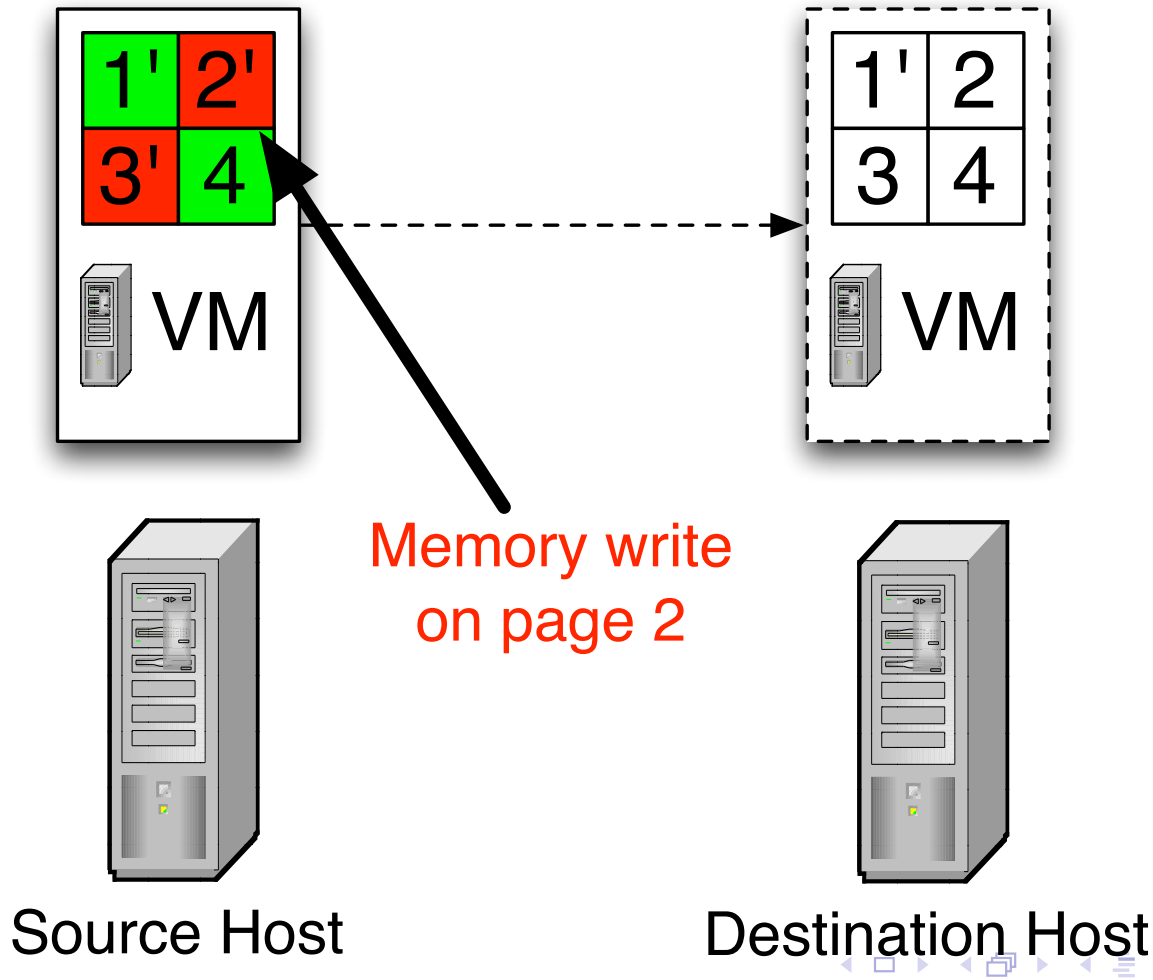
Pre-Copy Live Migration



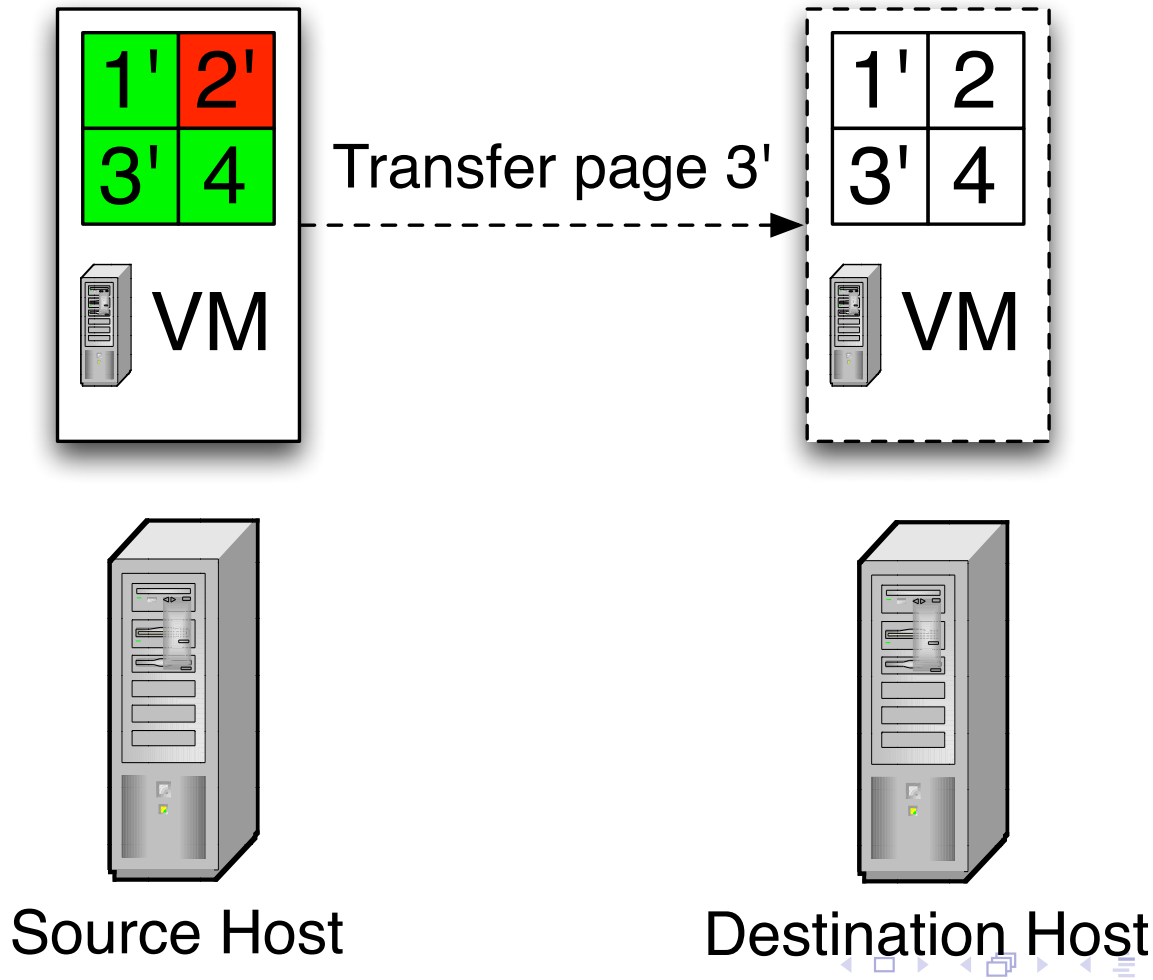
Pre-Copy Live Migration



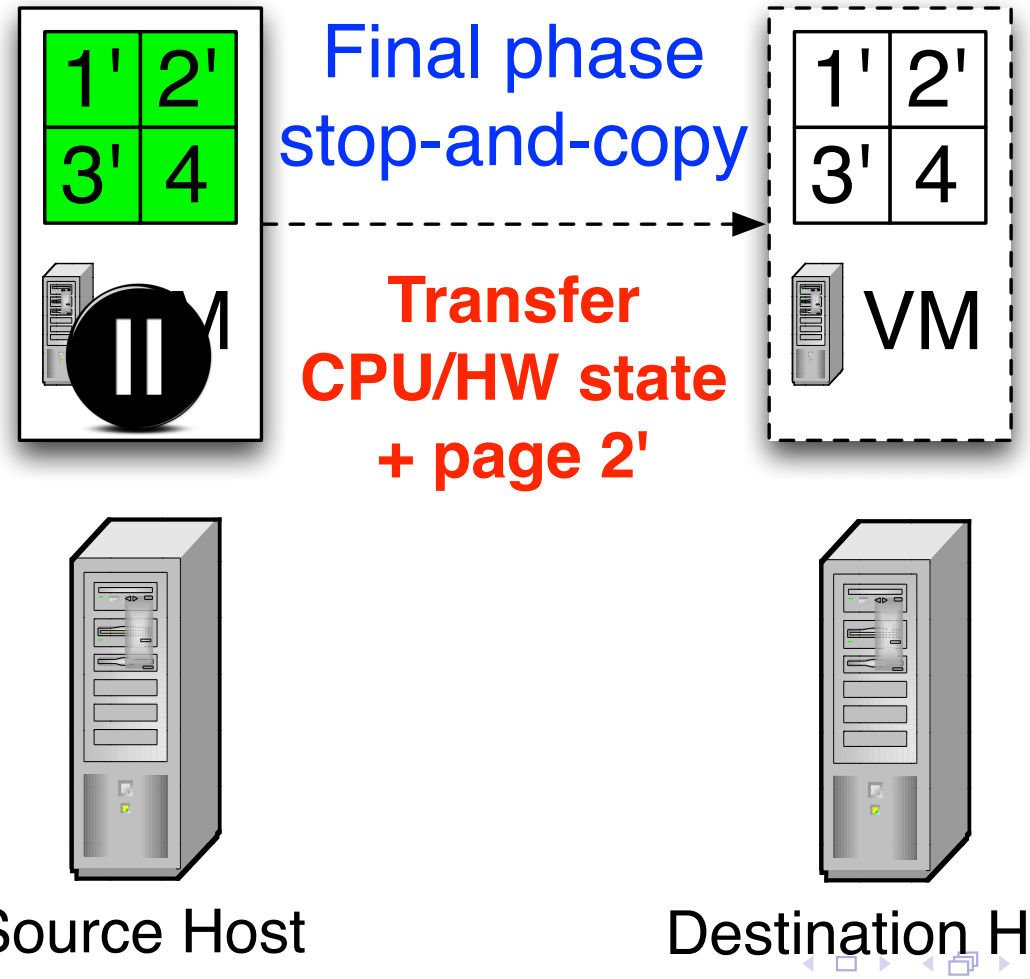
Pre-Copy Live Migration



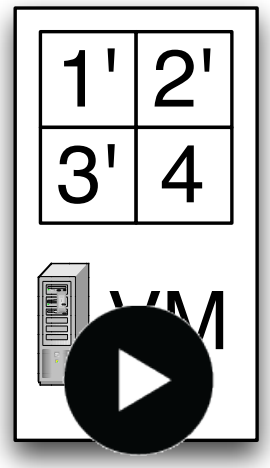
Pre-Copy Live Migration



Pre-Copy Live Migration



Pre-Copy Live Migration



Source Host

Destination Host

Stop-and-copy phase

- Different behavior between Xen and KVM
- Xen: threshold values
  - Remaining pages under a threshold
  - OR Too many iterations
  - OR Too much data transferred
- KVM: estimated downtime
  - Administrator can specify maximum downtime
  - Default: 30 milliseconds
  - KVM estimates available bandwidth
  - Stops only when estimated downtime $<$ maximum downtime
- Xen forces convergences of migration
- KVM trusts the admistrator or VM management software

Post-Copy Live Migration

- Pre-copy can present long downtime in the last phase
  - if the application modifies a large working set
  - if the available bandwidth is low
- Post-copy algorithm
  - Start by copying CPU and device state
  - Resume VM execution on the destination host
  - Fetch memory on demand when accessed
- Reduces downtime over pre-copy
- Can lower performance because of memory access latency
- KVM implementation: Takahiro Hirofuchi & Isaku Yamahata

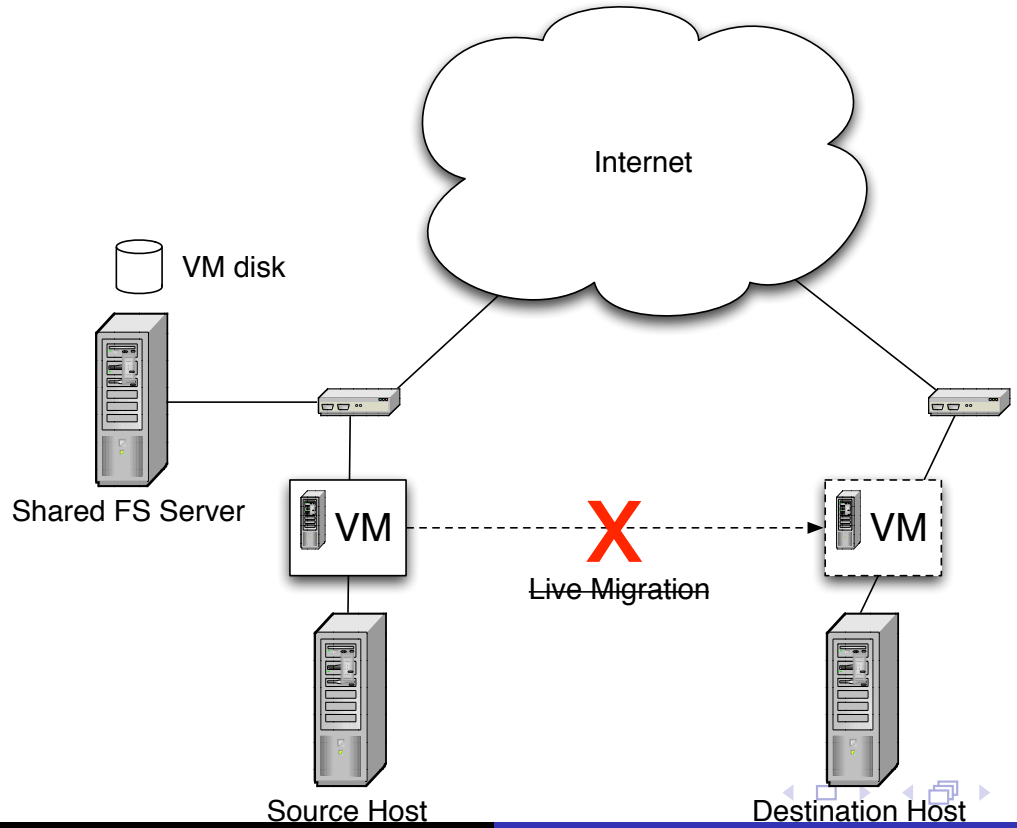
Trace & Replay Live Migration

- Use pre-copy as the basic migration algorithm
- Instead of sending modified memory pages $\rightarrow$ send external events of the VM to replay the modifications
- Example: network packet received
  - Log event and transfer to destination
  - Replay result of event on destination
- Greatly reduces amount of data to send between hosts
- Problem with SMP VMs as CPU synchronization would be too costly
- Liu et al., HPDC 2009

Outline

Live Migration over Wide Area Networks

- Live migration between different infrastructures/data centers/clouds



Live Migration of Storage

- Need to replicate data to the destination infrastructure
- Like memory migration, several algorithms exist
- Copy whole disk content + iteratively synchronize changes
  - KVM storage migration since 0.12
  - Developed by IBM in the RESERVOIR project [Nagin et al., SYSTOR 2011]
- Mirror writes to destination node
  - DRBD
  - Latest VMware ESX
- On-demand data fetching from destination
  - Hirofuchi et al., CCGrid 2009
  - KVM work-in-progress on image streaming (QED format)

Network Support for Live Migration

- Not possible to redirect traffic with ARP/RARP frames between different IP networks
- Various types of solutions
- Layer 2 VPNs
- Virtual networks based on reconfigurable overlays
- Mobile IP protocol
  - Home agent in the source network
  - Forwards to the foreign network of the mobile VM

Outline

Live Migration Optimizations

- Metrics to minimize
  - Total data transferred
  - Downtime
  - Total migration time
- Several approaches
  - Data Compression
  - Page Delta Transfer
  - Data Deduplication

Data Compression

- Compress memory pages sent over the network
- Compress zero'd memory pages → available in KVM
  - Interesting for migration of Windows
- Use a compression algorithm (gzip, bzip2, lzo)
  → KVM supports piping VM state to any executable
- Adaptive memory compression [Jin:2009]

Page Delta Transfer

- Memory pages are 4 KB on x86
- Modify 1 byte in the page $\rightarrow$ transfer 4 KB
- Delta transfer mechanism:
  - Keep copy of original page
  - Computer differences between original and new page
  - Send diff instead of full content
- Xor Binary Zero Run-Length-Encoding for KVM from SAP
  $\Rightarrow$ Live migration of large memory apps
- Discussion about support for KVM migration plugins

Data Deduplication

- VMs can contain identical data in multiple memory pages
- Remove duplicated memory pages
- Fast hash algorithm + full data comparison when match
  - Single-VM [Wood et al., VEE 2011]
  - Multi-VM on same host [Deshpande et al., HPDC 2011]
- Distributed approach for Multi-VM Multi-host
  [Riteau et al., Euro-Par 2011]

Outline

Conclusion

- Live migration still a hot topic of research
- Hypervisors start to include some advanced features
  - Storage migration
  - Optimizations
- Xen used to be the choice for hypervisor research
- Now KVM has a more dynamic community
- Ongoing & future research
  - Further live migration performance improvements
  - Especially in distributed systems & WANs
- Higher levels using live migration
  $\rightarrow$ Autonomous infrastructure management