Live Migration of Virtual Machines

Pre-copy: Christopher Clarke, Keir Fraser, et. al. NSDI 2005

Post-copy: Hines, Deshpande, Gopalan, VEE 2009
What is live migration?

• Move a VM from one physical machine to another even as its applications continue to execute during migration

• Live VM migration usually involves
  – Migrating memory state
  – Migrating CPU state
  – Optionally, migrating virtual disk state
Why Migrate VMs Live?

• Load Balancing

• System Maintenance

• Avoiding residual dependencies at source host which occurs with process migration
  – E.g. system call redirection, shared memory

• Avoiding Lost Connections
Performance Goals in Live Migration

• Minimizing Downtime

• Reducing total migration time

• Avoiding interference with normal system activity

• Minimizing network activity
Migrating Memory

- Pure stop-and-copy
  - Freeze VM at source,
  - Copy the VM’s pseudo-physical memory contents to target,
  - Restart VM at target
  - Long downtime.
  - Minimal total migration time = downtime

- Pure Demand Paging:
  - Freeze VM at source,
  - Copy minimal execution context to target
    - PC, Registers, non-pageable memory
  - Restart VM at target,
  - Pull memory contents form source as and when needed
  - Smaller downtime
  - Sloooow warm-up phase at target during page-faults across network
Pre-copy migration

- DON’T freeze VM at source ➔ Let it continue to run

- Copy VM’s pseudo-physical memory contents to target over multiple iterations
  - First iteration ➔ copy all pages.
  - Each subsequent iteration ➔ copy pages that were dirtied by the VM during the previous iteration

- Xend – a daemon in Domain 0 – maps the guest VM’s address space and transfers the pages over TCP connection to the target.

- Do a short stop-and-copy when number of dirty pages is “small enough”.

- But what if number of dirty pages never converges to a small enough number?
  - After a fixed number of iterations, give up and stop-and-copy.
Stages of Migration

1. Pre-Migration
   - Prepare the guest VM for migration via event channel notification
2. Reservation at target
   - Check if target has enough resources to receive the migrating VM
3. Iterative Pre-Copy
   - Copy memory contents over multiple rounds
4. Stop-and-Copy (downtime)
   - Freeze the guest and copy any residual state, including remaining dirty memory pages.
5. Commitment
   - Indicate to target machine that all state has been transferred
6. Activation
   - Target m/c restarts the guest
So what’s the catch?
How do we track dirtied pages?

- Mark the VM’s memory pages as read-only after each iteration.

- Trap write operations via hypervisor to xend and track dirtied pages.

- Reset after each iteration

- Works well as long as writes are infrequent
Optimizations

• Limit the bandwidth used by migration
  • To minimize impact on running services

• Stun Rogue Processes
  – Those that don’t stop dirtying memory

• Free Page Cache Pages
  – Can be re-cached at target
  – Potential performance hit
Post-copy migration

» Freeze the VM first
» Migrate CPU state and minimum state to destination
» Start VM at the target
  » but without its memory!

» Fetch memory by
  » Demand paging over network
  » Actively pushing from source

» Hopefully pages will be pushed BEFORE they are demand paged.
» Advantage: Each page transferred over the network only once.
» Disadvantage: Cold start penalty at the destination
Migrating Network Connections

- Migrating VM carries its
  - IP address,
  - MAC address, and
  - all protocol state, including any open sockets

- So nothing special to do while migrating within a switched LAN environment.

- What about the backward (re)learning delay at the network switches?
  - Switches needs to re-learn the new location of migrated VM’s MAC address
  - Solution: Send an unsolicited ARP reply from the target host.
  - Intermediate switches will re-learn automatically.
  - Few in-flight packets might get lost.
Storage Migration

• Much bigger problem
  – Many gigabytes of local disk image possible.

• Primarily an issue for migration over WAN

• One could bypass the problem for LANs
  – Assume the storage is over the network and remains accessible from the new target machine.
  – E.g. Network File System (NFS), or Network Block Device (NBD), or iSCSI etc.
Self Migration

• Guest OS migrates itself (mostly)

• Xend on source machine not involved.

• Migration stub needed at destination

• Challenge:
  – OS must continue to execute while transferring its final state.
  – Perform a careful (complicated) 2-stage checkpoint and copy.