

Xen Live Migration

Matúš Harvan

Networks and Distributed Systems Seminar, 24 April 2006

1 Xen Overview

2 Live migration

- General
- Memory, Network, Storage
- Migration Overview
- Writable Working Set
- Evaluation

- abstraction layer that decouples hardware
- full virtualization – guests are presented with a virtual machine identical to the real hardware
- paravirtualization – presenting a virtual machine abstraction, not identical to the underlying hardware, offering a special API
- several solutions/products available (VMware, User-Mode Linux, Xen)

- Xen Hypervisor – virtual machine monitor
- paravirtualization – provides idealized virtual machine abstraction
- allows for high performance virtualization
- operating systems have to be “ported” to Xen

- Xen hypervisor
 - only basic control operations, exported to authorized domains
 - complex policy decisions performed in a guest OS in a domain
- domain0
 - special management domain with additional privileges
 - device drivers
 - creates and manages domUs
- domU
 - guest
 - domUs access only simple, virtualized hardware abstractions
 - possible to give a domU access to specific PCI devices so that a device driver would run in a domU rather than domain0

- CPU level protection
 - using x86 CPU rings
 - CPU ensures that the domains have to use the Xen hypervisor to do privileged operations
- memory
 - memory allocations from a reserved pool
 - page table updates validated by hypervisor
- Device I/O
 - real hardware handled by device drivers in domain0
 - Xen exports a simple abstraction interface for domUs to use devices
 - I/O data transferred using shared-memory, asynchronous buffer-descriptor rings
 - interrupts replaced with a lightweight event-delivery mechanism

- *Virtual Network Interfaces*
 - creation and deletion validated by Xen
 - filtering rules - changes validated by Xen
 - prevent source address spoofing
- *Virtual Block Device*
 - associated access-control information (which domain) and restrictions (i.e. read-only)

- migrating whole OS with running applications (kernel-internal state and application-level) rather than single processes
- avoid *residual dependencies*
- clusters, data centers
- load balancing, hardware maintenance
- separation of concern between users and operators (no need for operators to access domUs)

- minimize
 - *downtime* – service unavailable as no currently executing VM available
 - *total migration time* – duration from migration initiation to being able to discard original VM
 - disruption of active services through resource contention (CPU, network bandwidth)
- virtual machine – encapsulates access to HW
 - memory
 - network
 - (disk) storage

- Possible approaches for memory transfer:
 - **Push phase** – The source VM continues running while certain pages are pushed across the network to the new destination. To ensure consistency, pages modified during this process must be re-sent.
 - **Stop-and-copy phase** – The source VM is stopped, pages are copied across to the destination VM, then the new VM is started.
 - **Pull phase** – The new VM executes and, if it accesses a page that has not yet been copied, this page is faulted in (pulled) across the network from the source VM.

- Xen uses a *pre-copy phase* iteratively copying modified pages in rounds, then stops VM and copies remaining memory pages (*stop-and-copy phase*)
- dynamic rate-limiting algorithm to minimize resource contention and decide when to end pre-copy phase and enter stop-and-copy phase

- maintain all open network connections without relying on forwarding mechanisms on the original host
- migrated VM keeps protocol state and IP address, advertising that the IP address has moved
- generate an unsolicited ARP reply after migration
 - some routers do not accept broadcast ARP replies
 - can send directed replies only to entries in its own ARP cache
 - alternatively, keep original MAC address, network switch has to figure out it moved to a different port
- works only within a single switched LAN
- wide-area network migration not supported

- local-disk storage not migrated
- need network-attached storage (NAS), e.g. iSCSI
 - uniformly accessible from all hosts in the cluster
- alternative for experimenting: ram disk

- at any time at least one host has a consistent VM image
- migration process is viewed as a transactional interaction between two hosts
- VM not exposed more to system failure than when running on original single host

Migration Overview

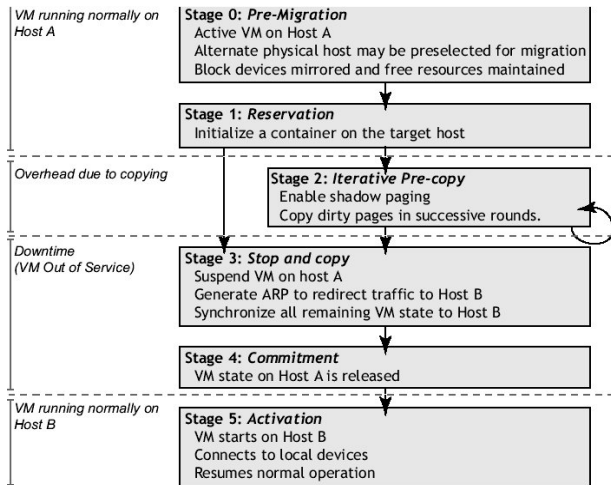


Figure: Migration timeline [2]

Writable Working Set

- pre-copy migration of not so often modified pages
- rapidly modified pages get transferred after VM is stopped
- **writable working set** (WWS) – pages modified so often that it does not make sense to move them during pre-copy phase
- using Xen's shadow page tables to trace WWS
- performed several experiments with different benchmarks to track WWS
- conclusions:
 - pre-copy migration performs better than a naive stop-and-copy
 - decreased downtime, but diminishing returns w.r.t. increasing number of iterations
 - hottest pages dirtied faster than transferred – limit on minimum possible downtime
 - makes sense to increase bandwidth limit for later (and shorter) rounds

Dynamic Rate-Limiting

- dynamically adapt the bandwidth limit during each pre-copying round
- administrator selects a minimum and maximum
- first round uses minimum bandwidth
- further rounds increase bandwidth limit by certain amount (50Mbit/sec empirically found)
- terminate pre-copying when calculated bandwidth limit greater than maximum or less than 256KB remains to transfer
- final stop-and-copy uses maximum bandwidth for memory transfer to minimize downtime
- bandwidth remains low while transferring most pages, increasing only at the end for the “hottest” pages – balances short downtime with low average network contention and CPU usage
- rapid page dirtying – page transferred if modified during previous round, but not during this round (otherwise likely to be modified again)

- Stunning Rogue Processes
 - monitor WWS of individual processes
 - limit each process to 40 write faults
- Freeing Page Cache Pages
 - return free and cold buffer pages to Xen rather than migrating them

Evaluation – Simple Web Server

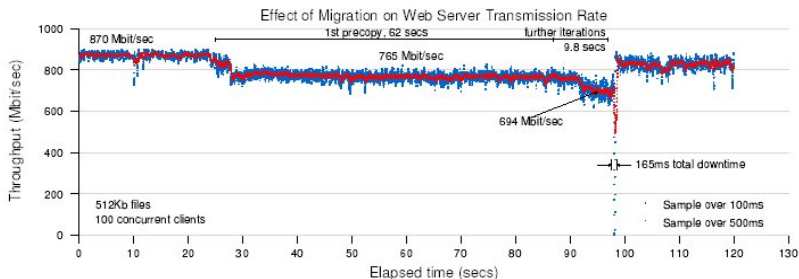


Figure: Migrating a running web server VM[2]

- Apache 1.3 web server serving a static 512KB file
- 165ms downtime

Evaluation – Complex Web Workload: SPECweb99

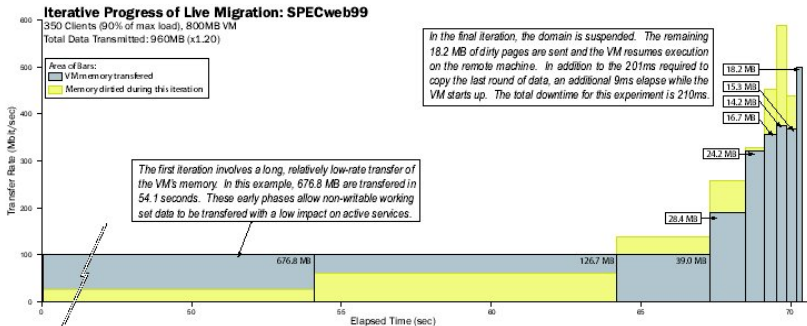


Figure: Migrating a running SPECweb VM[2]

- running at 90% of maximum load
- 210ms downtime
- no decrease in number of conformant clients

Evaluation – Low-Latency Server: Quake 3

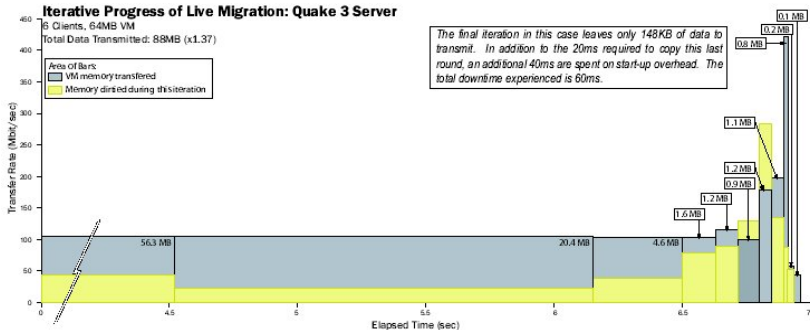


Figure: Migrating a running Quake3 VM[2]

- VM with 64MB of memory, network game with six players
- downtime of 60ms, observed as a 50ms increase in response time
- not noticed by players

Evaluation – A Diabolic Workload: MMuncher

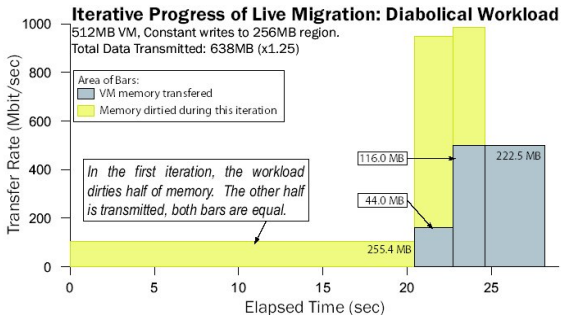




Figure: Migrating a VM running a diabolical workload[2]

- VM with 512MB of memory, C program constantly writing to a 256MB memory region
- downtime of 3.5s
- rare in real workloads

- Cluster Management
 - develop cluster control software capable of making informed decisions for placement and movement of VMs
- Wide Area Network Redirection
 - layer 2 redirection not possible outside a local subnet
 - use Mobile IP
 - connection migration at TCP level
 - use Dynamic DNS to locate host after move
- Migrating Block Devices
 - total migration time would be significantly extended if transferring a complete local disk
 - mirroring disk contents on remote hosts – RAID system across several machines, multiple hosts acting as storage target for one another
 - copy-on-write filesystems

- live OS migration with Xen
- SPECweb99 – 210ms downtime
- Quake3 – 60ms downtime
- dynamic bandwidth adaptation minimizes impact on running services while minimizing total downtime below discernable thresholds
- suitable for well-connected data-center or cluster with network-accessed storage

-  P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield
Xen and the Art of Virtualization
In Proceedings of the 19th ACM Symposium on Operating Systems Principles, October 2003.
-  C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Kimpach, I. Pratt, and W. Warfield
Live Migration of Virtual Machines
In 2nd USENIX Symposium on Networked Systems, Design and Implementation (NSDI 05), pages 273286, May 2005.