Minimum-energy bandwidth management for QoS live migration of virtual machines

D. Amendola; N. Cordeschi; E. Baccarelli
broadcom.diet.uniroma1.it/People/Danilo-Amendola - www.daniloamendola.it
danilo.amendola@uniroma1.it - appeared on Computer Networks - 2015,

2nd IEEE ComSoc Summer School
19-23 June '16 - Trento (Italy)

Abstract

Live virtual machine (VM) migration aims at enabling the dynamic balanced use of the networking/computing physical resources of virtualized datacenters, so to lead to reduced energy consumption. However, the bandwidth consumption and latency of current state-of-the-art live VM migration techniques still reduce the experienced benefits to much less than their potential. Motivated by this consideration, in this paper, we analytically characterize, prototype in software and test the optimal bandwidth manager for intra-datacenter live migration of VMs. The goal is the minimization of the migration-induced communication energy under service level agreement (SLA)-induced hard constraints on the total migration time, downtime, slowdown of the migrating applications and overall available bandwidth. For this purpose, after recognizing that the resulting (nonconvex) optimization problem is an instance of Geometric Programming, we solve it by resorting to suitably developed adaptive version of the so-called primal-dual gradient-based iterations and, then, we analytically characterize its feasibility conditions.

Introduction and Architecture

Virtualization is an emerging technique that allows running multiple operating systems (OSs) simultaneously on a single server. A special middleware layer, the virtual machine manager (VMM) or hypervisor, abstracts from physical computing/networking resources. In modern virtualized networked datacenters (VNetDCs), live migration allows to move a continuously running VM from one server to another, so to attain multiple goals, including failure tolerance and energy-saving through server consolidation/load balancing. Although live migration is becoming a service primitive function for the resource management of VNetDCs, it may induce slowdown of the application, as well as not negligible increments of the networking traffic and the computing-plus-networking energy consumption.

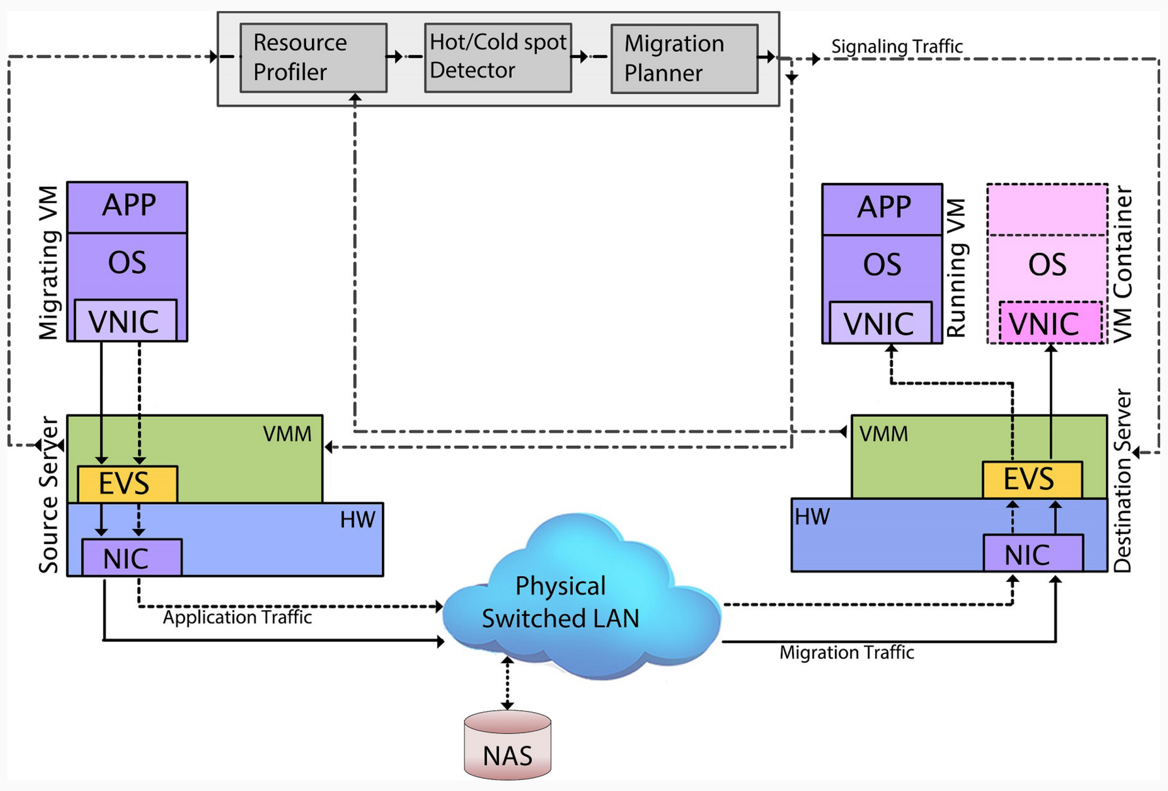
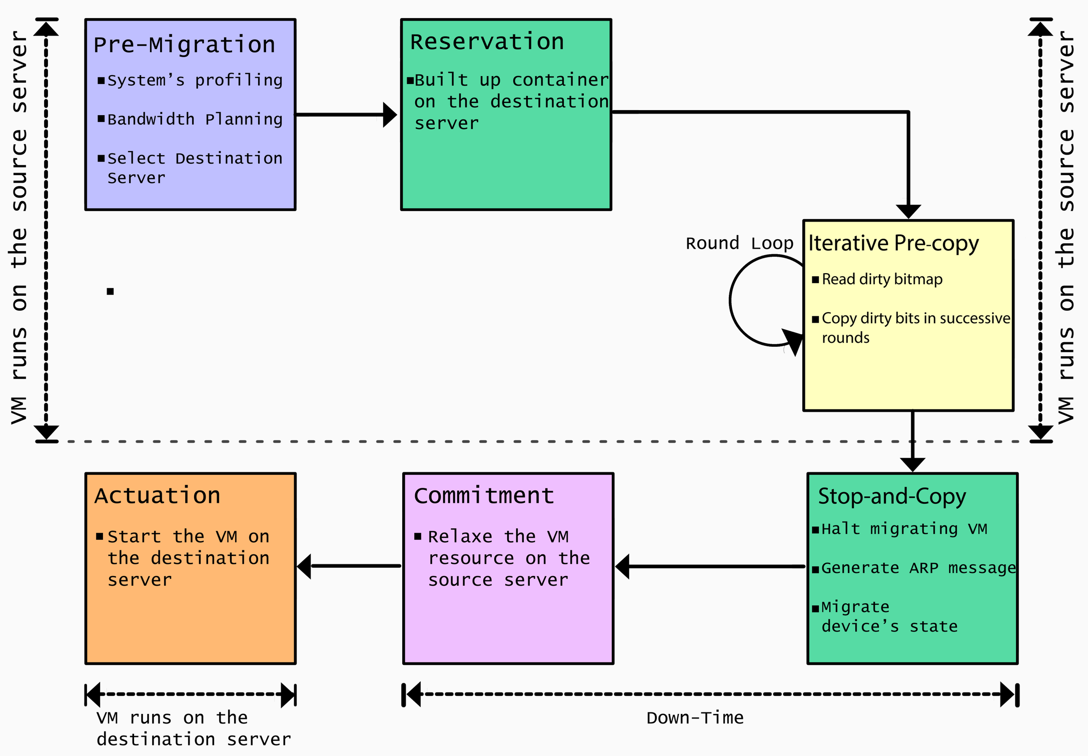


Fig. 1: Reference VNetDC architecture for intra-datacenter live VM migration.

In Fig.1: Continue/dotted arrowed lines denote end-to-end TCP connections conveying migration/application) traffic. Dashed-dotted arrowed lines denote signalling flows for implementing the resource profiling and migration plan. APP: application; VMM: virtual machine manager; OS: operating system; HW: networking/computing hardware; EVS: external virtual switch; VNIC: virtual network interface card; NIC: physical network interface card; NAS: network-attached storage.

Pre-copy live VM migration



The PeCM technique involves six stages:

1. *Pre-migration:* resource profiling. (T_{PM} seconds;)
2. *Reservation:* the computing/communication/storage/memory physical resources are reserved at the destination server. (T_{RE} seconds);
3. *Iterative pre-copy:* First, entire memory of the migrating VM is sent to the destination server. Hence, the memory pages modified during the previous copy-round are re-transferred to the destination server (see Fig. 3), (T_{IP} seconds);
4. *Stop-and-copy:* the migrating VM is halted and a final memory-copy round is performed (see Fig. 3). (T_{SC} seconds);
5. *Commitment:* Destination server notifies that it has received successfully. (T_{CM} seconds);
6. *Re-activation:* the I/O resources and IP address are re-attached to the VM on the destination server. (T_{AT} seconds).

Problem setup and optimal resource allocation

Iterative Pre-Copy Migration (PeCM) technique. R (Mb/s) is the transmission rate.

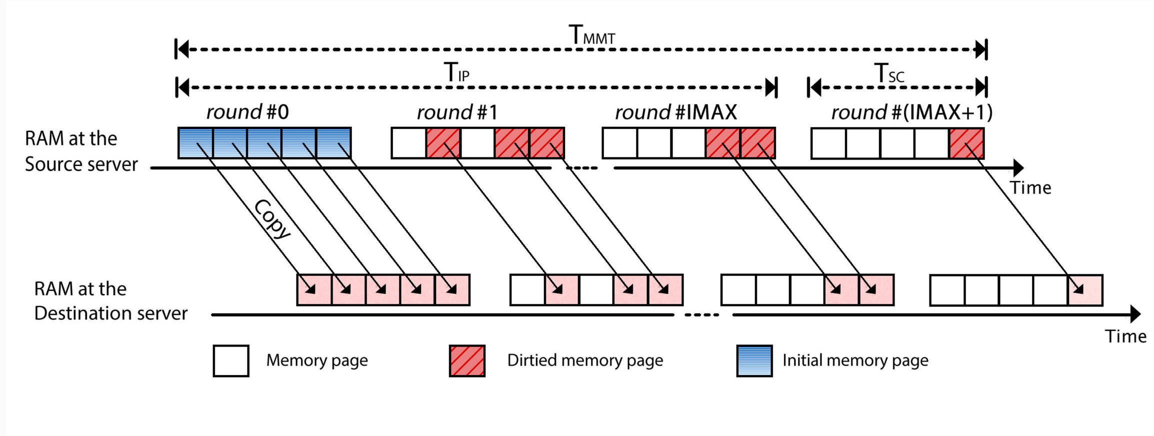


Fig. 3: Time-chart of the PeCM technique.

The total communication energy $\mathcal{E}_{TOT}$ (J) consumed during the migration process:

$$\mathcal{E}_{TOT} \equiv \mathcal{E}_{TOT}(R) = \left\{ K_0 M_0 R^{\alpha-1} \left[1 + \theta \left(\sum_{i=1}^{I_{MAX}+1} (\overline{w}/R)^i \right) \right] \right\} + \mathcal{E}_{SETUP}, \quad (1)$$

The considered QoS Bandwidth Manager Optimization Problem is formally defined as in:

$$\min_{R \geq 0} \; \mathcal{E}_{TOT}(R),$$

s.t.: constraints in (2), (3), (4) *and* (5) .

$$\Psi_1(R) \triangleq \theta[(T_{MMT}(R)/\Delta_{MMT}) - 1] \leq 0, \quad (2)$$

$$\Psi_2(R) \triangleq [(T_{SC}(R)/\Delta_{SC}) - 1] \leq 0. \quad (3)$$

$$\Psi_3(R) \triangleq \theta[(\beta\overline{w}/R) - 1] \leq 0. \quad (4)$$

$$(R/\widehat{R}) - 1 \leq 0, \quad (5)$$

$$\widehat{R} \triangleq \{R_{MAX};\ \rho_{MAX} R_{TOT}\},$$

Tracking capabilities under connection phenomena

How our solution perform under variable connection phenomena:

- left) Case of time-varying dirty pages rate: $\overline{w}$
- right) Case of time-varying connection quality: K_0 (is the summerized value that model the overall effect on the performed migration of the aggregate traffic supported by the datacenter connection).

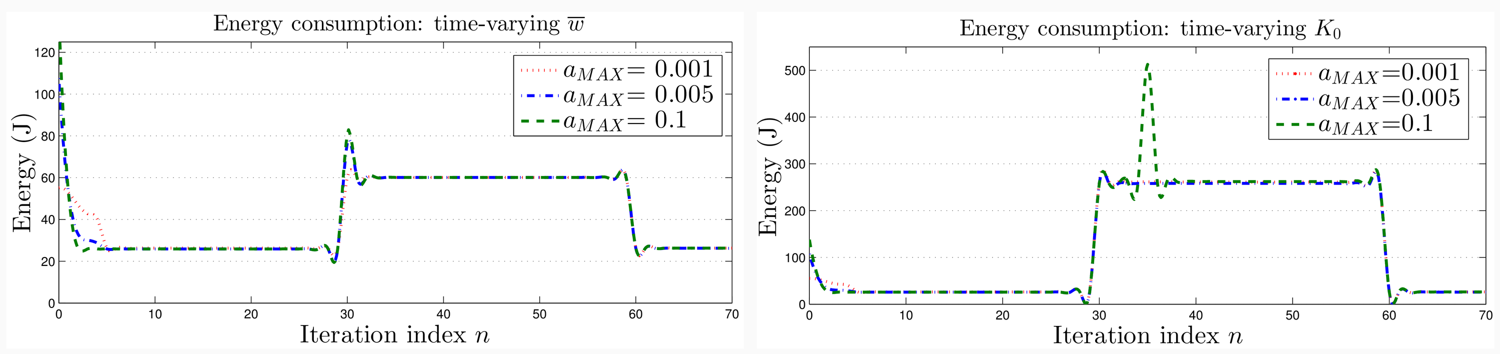


Fig. 4: Time evolutions (in the n index) of the energy consumption of the proposed bandwidth manager at: $\widehat{R} = 900$ (Mb/s), $M_0 = 512$ (Mb), $\beta = 1.15$, $\Delta_{MMT} = 13.5$ (s), $\Delta_{SC} = 0.6$ (s), $\widetilde{I}_{MAX} = 3$, and $\gamma = 100$. (a) Case of time-varying $\overline{w}$; (b) Case of time-varying K_0.

I_{max} formula, validation results

We claim that the optimized setting: $\widetilde{I}_{MAX}$ of I_{MAX} is obtained by computing the value with the equality:

$$\widetilde{I}_{MAX} \equiv \left\lceil \frac{\log(M_0/\Delta_{SC}\widehat{R})}{\log(\widehat{R}/\overline{w})} - 1 \right\rceil, \text{ for } (\widehat{R}/\overline{w}) > 1, \quad (6)$$

We validated this formula with estensive simulations, here we present some results:

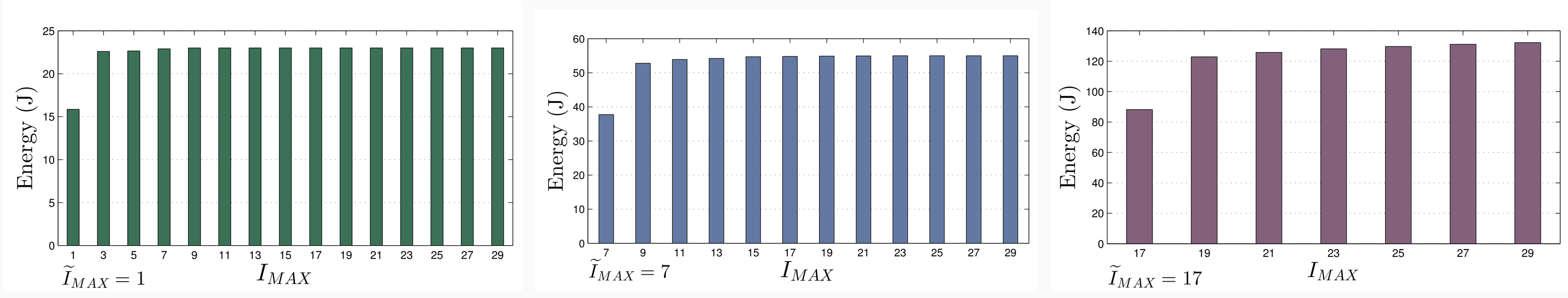


Fig. 5: $\mathcal{E}^*_{TOT}$ - vs. - I_{MAX} behavior for the proposed bandwidth manager at: $\widehat{R} = 100$ (Mb/s), $M_0 = 512$ (Mb), $\beta = 1.10$, $\Delta_{MMT} = 50$ (s), $\Delta_{SC} = 0.3$ (s). The application scenario is considered at: (left) $\overline{w} = 60$ (Mb/s); (center) $\overline{w} = 75$ (Mb/s); and, (roght) $\overline{w} = 90$ (Mb/s).

Conclusion We developed the optimal bandwidth manager for intra-datacenter live VM migration. It minimizes at run-time the communication energy wasted by the migration of the VM memory under hard QoS constraints on both the migration time and downtime. After implementing it atop a wired test-bed, we measured and compared its energy performance by considering synthetic and real-world workloads, as well as random and ordered migration scheduling disciplines. The carried out field trials highlight that the average energy saving of the proposed bandwidth manager over the corresponding state-of-the-art Xen one is over 40% and approaches 66% under strict constraints on the tolerated downtimes. Interestingly, the measured per-migration CPU slow-down induced by its implementation is, in average, limited up to 1.5–2%, while the measured average stretching of the execution times of the migrated applications is under 20%.