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

D. Amendola; N. Cordeschi; E. Baccarelli
danilo.amendola@uniroma1.it - Computer Networks - 2015, poster presentation at ComSoc Summer School '16

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 through field trials 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. Hence, we prototype the resulting bandwidth manager atop an intra-datacenter wired test-bed, and, then, test and compare its energy performance through extensive field trials. The carried out field trials point out that: (i) the *energy savings* attained by the proposed bandwidth manager over the state-of-the-art ones currently utilized by Xen, KVM and VMware hypervisors are *over* 40% and *approach* 66% under strict QoS constraints; (ii) the proposed bandwidth manager is capable to *quickly adapt* to the abrupt changes possibly experienced by the dirty rates of the running applications and/or the round trip times of the utilized (possibly, congested) TCP/IP connections; and, (iii) its actual implementation may be carried out in a *distributed* and *scalable* way, and it consumes *less* than 1.5% of the CPU computing power per migrated VM.

Introduction and Architecture

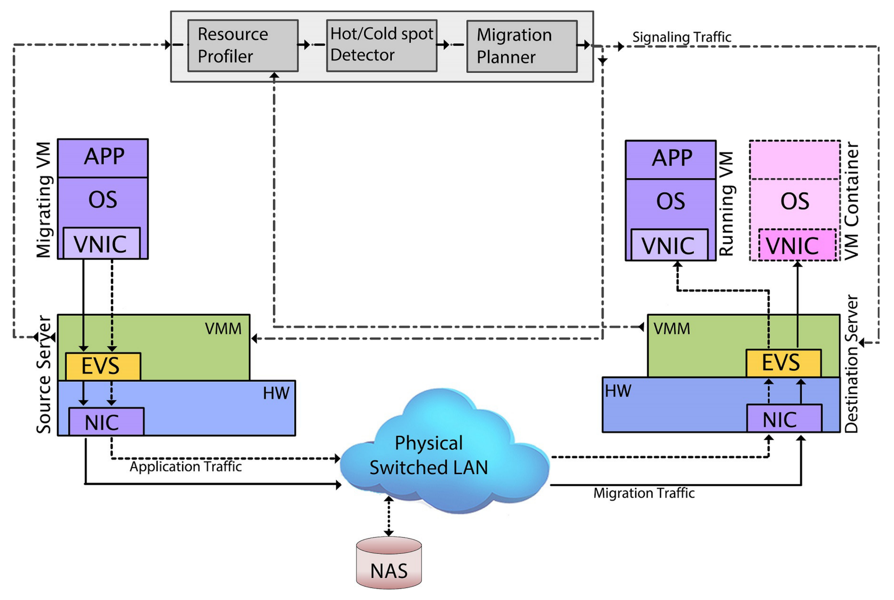


Fig. 1: Reference VNetDC architecture for intra-datacenter live VM migration. Continue (resp., dotted) arrowed lines denote end-to-end TCP connections conveying migration (resp., application) traffic. Dashed-dotted arrowed lines denote signalling flows for implementing the resource profiling and migration 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.

Problem setup and optimal resource allocation

Let R (Mb/s) be the transmission rate used during the third and fourth stages for migrating the VM, that is, the migration bandwidth. Since, by definition, only T_{IP} and T_{SC} depend

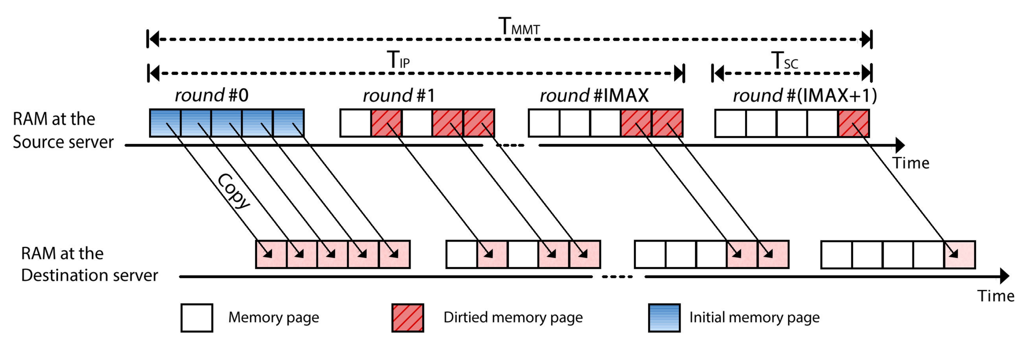


Fig. 2: Time-chart of the PeCM technique.

on R, while all the remaining migration times in Eqs. (??) and (??) play the role of constant parameters, in the sequel, we focus on the evaluation of the (already defined) stop-and-copy time T_{SC} and the resulting memory migration time T_{MMT}, which is defined as in:

$$T_{MMT} \equiv T_{MMT}(R) \triangleq T_{IP}(R) + T_{SC}(R). \quad (1)$$

BroadcomLab Research Group

Keyword: Mobile Cloud Computing; Vehicular Cloud Computing; Big Data; Green Data Center; Communication and Computing Engineering in Cloud and Big Data; LTE; Next Generation ADSL.

BroadcomLab
DIET Dept. at Sapienza University of Rome
First floor, room 101 - Via Eudossiana, 18, Rome (Italy)
phone: 0039 06 4458 5 366 - *broadcom.diet.uniroma1.it*

Results on the tracking capabilities under connection

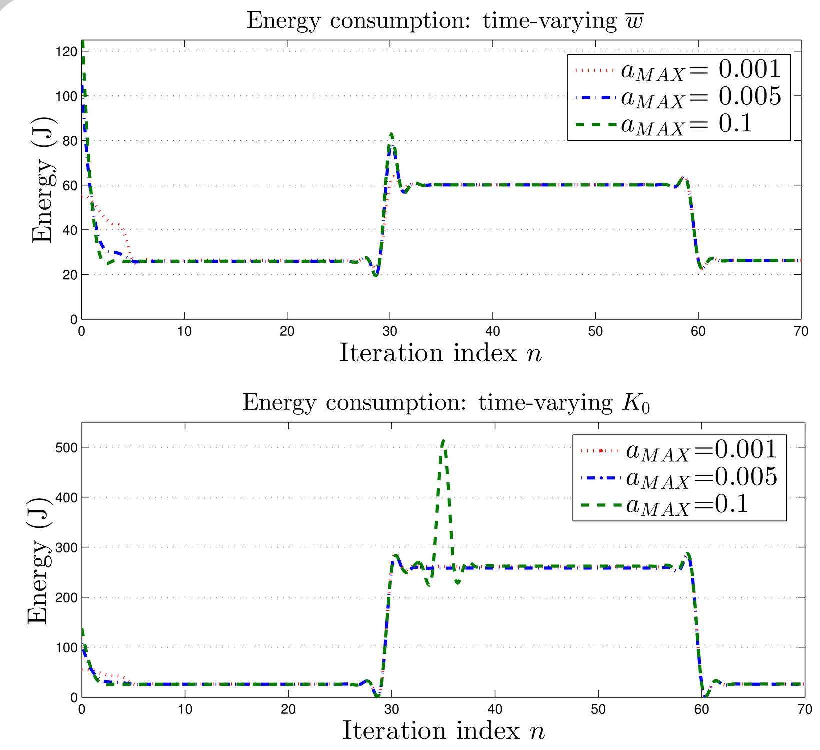
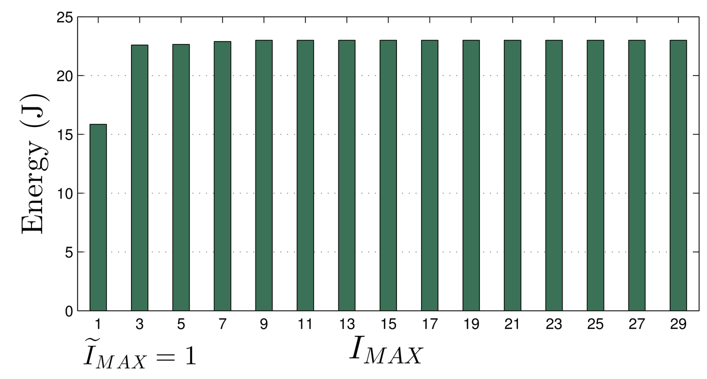


Fig. 4: Time evolutions (in the n index) of the energy consumption of the proposed bandwidth manager at: $\widehat{R} = 900$ (Mb/s), $M_0 = 512$ (Mb), $\beta = 1.15$, $\Delta_{MMT} = 13.5$ (s), $\Delta_{SC} = 0.6$ (s), $\widetilde{I}_{MAX} = 3$, and $\gamma = 100$ for the application scenario of Section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

Results Imax

To evaluate the effect on $\mathcal{E}^*_{tot}$ of the networking powers, we have set: $T_t = 500$ (ms), $C_{max} = 150$ $(Mbit/s)$, $L_t = 10$ $(Mbit)$, $k_e = 0.05(mJ)/(MHz)^2$, $f_i^{max} = 100$ $(Mbit/s)$, $f_i^0 = 0$ $(Mbit/s)$, $\mathcal{E}_i^{max} = 1$ (mJ), $\Delta(i) = 1$ (s). We have evaluated $\mathcal{E}^*_{tot}$ $(Joule)$ for the following three network scenarios: *i)* $P_i^{net} = 0$ (mW) (i.e., no communication costs); *ii)* $P_i^{net} = 1$ (mW) (i.e., homogeneous communication costs); and, *iii)* $P_i^{net} = 1 + 0.25(i-1)$ (mW), for $i = 1, \ldots, M$ (i.e., heterogeneous communication costs).



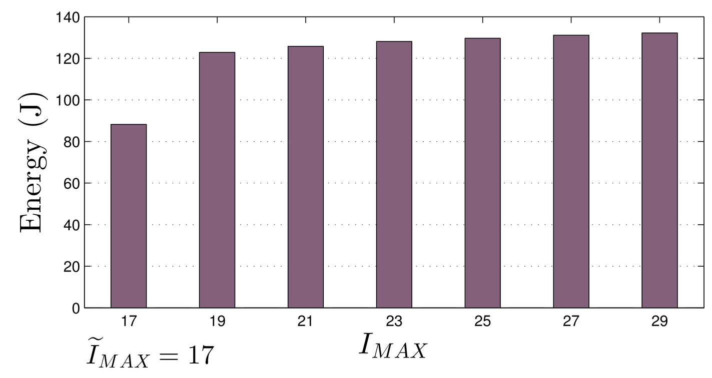


Fig. 6: ...

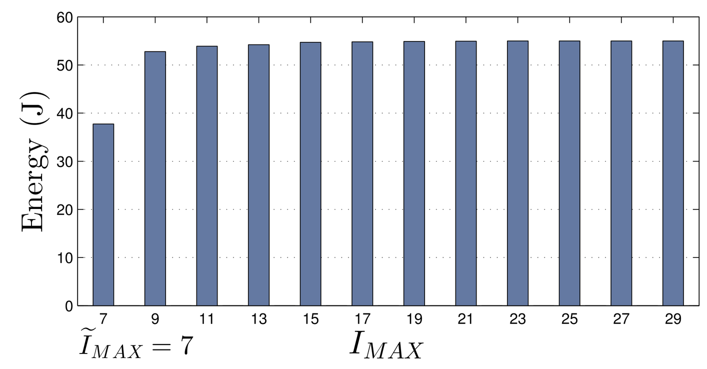


Fig. 5: Time evolutions (in the n index) of the energy consumption of the proposed bandwidth manager at: $\widehat{R} = 900$ (Mb/s), $M_0 = 512$ (Mb), $\beta = 1.15$, $\Delta_{MMT} = 13.5$ (s), $\Delta_{SC} = 0.6$ (s), $\widetilde{I}_{MAX} = 3$, and $\gamma = 100$ for the application scenario of Section ??. (a) Case of time-varying $\overline{w}$; (b) Case of time-varying K_0.

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%.