Does Live Migration of Virtual Machines cost Energy?

Anja Strunk and Waltenegus Dargie
Chair of Computer Networks
Faculty of Computer Science
Technical University of Dresden
01062 Dresden, Germany
Email: {anja.strunk, waltenegus.dargie}@tu-dresden.de

***Abstract*—Live migration, the process of moving a virtual machine (VM) interruption-free between physical hosts is a core concept in modern data centers. Power management strategies use live migration to consolidate services in a cluster environment and switch off underutilized machines to save power. However, most migration models do not consider the energy cost of migration. This paper experimentally investigates the power consumption and the duration of virtual machine migration. We use the KVM platform for our experiment and show that live migration entails an energy overhead and the size of this overhead varies with the size of the virtual machine and the available network bandwidth.**

I. Introduction

Virtualization is a technique that enables several operating systems to run simultaneously on a single physical machine. It has become a core aspect in modern servers and data centers due to several advantages, such as flexible and efficient sharing of resources, fault tolerance, portability, and cost efficiency [1]. In a virtualized environment, virtual machines (VM) acting like real physical machines can run in parallel and in isolation from each other and yet sharing the same physical resources. A low level middleware called a hypervisor abstracts these virtual machines from the physical hardware and determines the exclusive use of resources by each VM. The structure of a virtualized computing environment is displayed in Figure 1.

One of the key features of virtualization is the live migration of virtual machines which enables an active (executing) VM to be moved from one physical machine to another in a transparent fashion [2]. This key feature has become a significant tool for a variety of scenarios. Some of which include:

- *Load balancing* [3], [4]. The aim is to adjust a virtual machine placement in order to achieve critical business goals, such as high throughput.

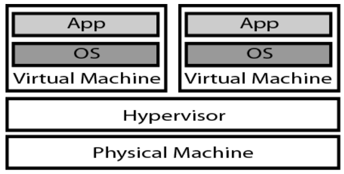

Fig. 1. The principle of virtualization.

- *Transparent IT maintenance*. Administrators can transparently move virtual machines to free and shut down hosts for maintenance.
- *Power management* [5]–[7]. The aim is to consolidate virtual machines through live migration on an optimal number of servers and to switch off underutilized servers. The optimality criteria is the minimization of the energy consumption of the data center [8], [9].

Although live migration is widely used by the industry as well as the research community, most existing or proposed approaches disregard the cost of migration. For example, the live migration approach of Li et al. for energy-saving application placement in cloud computing environment disregards the cost of migration [10]. The same view is shared by similar approaches [11]–[13]. However, so far there is no theoretical or experimental proof that suggest the live migration of virtual machine is for free.

This paper experimentally investigates the migration cost from an energy point of view. We will show that live migration has indeed an energy overhead and the size of this overhead depends on factors such as the size of the VM and the available bandwidth in the network.

The rest of the paper is organized as follows. In Section II we explain the technical aspect of live migration. In Section III, we summarize related work. In Section IV, we analyze the energy overhead due to live migration of virtual machines. Finally, in Section V, we outline some open research issues in this area and give concluding remarks.

II. Virtual Machine Live Migration

Live migration enables a virtual machine to be physically moved from one physical host to another, in a transparent fashion, while the virtual machine is still running. The current virtualization technology (based on hypervisors) does not use local discs to store VM images. Instead it requires a network attached storage (NAS) that can be accessible to all hosts and serve as hard drive for the virtual machines. By using a NAS, the process of live migration is limited to copying the in-memory state and the content of the CPU registers between the physical machines. For this task modern virtualization systems use a technique called pre-copy [2], consisting of the following three phases (see Figure 2):

1) *Pre-Copy Phase*: At this stage, the VM continues to run while its memory is iteratively copied page-wise from the source machine to the destination host. Iteratively means the algorithm works in several rounds. It starts with transferring all active memory pages. As each round takes some amount of time, some of the memory pages on the source machine may be changed (dirtied) and may no longer be in sync with the copy version on the destination host. These pages have to be re-sent to ensure memory consistency.
2) *Pre-Copy Termination Phase*: Without any stop condition, the iteratively pre-copy phase may carry on indefinitely. Stop conditions depend on the design of the hypervisor, but typically, they take one of the following thresholds into account: (1) the number of iterations exceeds a pre-defined threshold ($n > n_{th}$), (2) the total amount of memory that has already been transmitted exceeds a pre-defined threshold ($mem_{mig} > mem_{th}$), or (3) the number of pages dirtied in the previous round falls below a pre-defined threshold ($pg < pg_{th}$).
3) *Stop-and-Copy Phase*: At this stage the hypervisor suspends the VM to prevent further page writing and copies the remaining dirty pages as well as the state of the CPU registers to the destination host. After the migration process is completed, the hypervisor on the destination host resumes the VM.

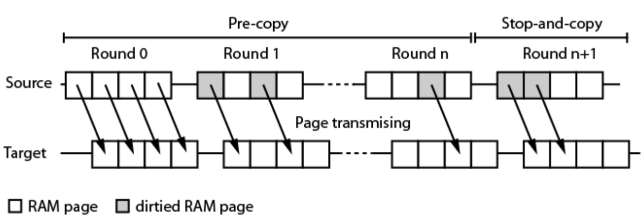

Fig. 2. Live Migration algorithm performs memory transfer pagewise in several rounds [14].

III. Related Work

Live migration has been investigated in various contexts [5], [15]–[20]. Most of the existing or proposed approaches focus on the performance of live migration and measure migration time and down time, under different conditions. Work that explicitly investigates the costs of migration is rare.

We classify the costs of virtual machine live migration into performance loss and energy overhead. During live migration, a hypervisor labels all memory pages occupied by a VM as read-only in order to facilitate migration. All requests to overwrite some of these pages will raise exception and handled by the hypervisor. This slows down the VM's response to requests and reduces its throughput [2]. Additional performance loss arises due to resource bottlenecks. The pre-copy and stop-and-copy processes require additional resources, particularly, network bandwidth and some CPU cycles. Since co-located virtual machines must not be affected by the migration, there may be a resource deficiency for the VM being migrated [21].

Kuno et al. investigate the processing speed of CPU-intensive and the reading speed of IO-intensive (disk) workloads. The authors find out that the performance of CPU-intensive workloads decelerates by 15% whereas the reading speed diminishes by 10% [22]. In [2], the authors demonstrate that the transmission rate of an Apache Web Server slows down by 12 to 20%. Performance loss may be problematic in systems where the response time constitutes a strict performance guarantee. For example, Voorslys et al. show that 90% of the download time of home pages created with Web 2.0 technologies (PHP, Ruby on Rails, J2EE) may not be accessible during live migration [23].

The additional resource utilization cost during live migration creates an energy overhead. However, current live migration scenarios do not consider this energy cost. For example, Mistral proposes a framework to optimize the power consumption of cloud systems and uses live migration as a mechanism to consolidate virtual servers and switch off underutilized physical machines. The framework does not take the migration's additional power consumption into account [24]. This idea is shared by similar approaches which investigate service consolidation and dynamic power management in data centers [13], [25]–[27].

This paper complements the study which focuses on the performance cost of migration and addresses the energy cost during the live migration of virtual machines. We will show how this cost varies as the size of the virtual machine and the available network bandwidth varies.

IV. Experiments

The central research question we would like to address can be formulated as follow:

1) How much is the energy overhead during live migration?
2) Which parameters influence the energy overhead of a live migration?

The average Energy, E, of a migration is defined as the average power, P, multiplied by the migration duration τ:

$$E = P \times \tau \tag{1}$$

To quantify the energy cost of a VM migration, we measure the power consumption of the source and destination servers before and during migration and record the migration duration. To isolate the cost of migration from all other costs due to uncontrolled activities, we carry our migration when both servers are idle with 0% CPU utilisation. The parameters we choose to closely investigate are the VM's memory size and available transmission bandwidth at the source and destination servers, as the migration time strongly varies with these parameters [14], [16].

A. Cluster Set up

The server cluster we set up for our experiment consists of two identically constructed servers (we call them Gandalf and Wuotan), a client server, and a network attached storage (NAS). All devices are connected with each other via a 1

GBit/s switch (Figure 3). The servers run Fedora 15 [28] (Linux kernel v. 2.6.38, x86_64) in which KVM is used as a hypervisor. We use the open source operating system FreeNAS [29] (v. 8.0.1, AMD_64) as a Network Attached Storage.

Each server employs an Intel I5-680 Dual Core 3.6 MHz processor, 4 GB DDR3-1333 SDRAM memory and a 1 Gbit/s Ethernet Network Interface Card (NIC). The NAS system is equipped with one Intel Xeon E5620 Quad-Core 2.4 MHz processor, 10 GB DDR3-1333 SDRAM memory and 1 Gbit/s Ethernet NIC.

Athena, the virtual machine under test, runs Fedora 15 (Linux kernel v. 2.6.38, x86_64) with one virtual processor and variable memory size and network bandwidth. The client, written in C and hosted on a third physical machine, triggers the live migration of Athena from Wuotan to Gandalf using libvirt [30], a toolkit enabling interactions with the hypervisor and the operation system.

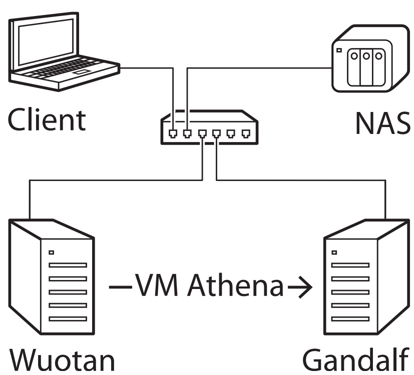

Fig. 3. The set up for migration cluster.

B. Measurements

We run for each parameter setting 25 iterations. Each iteration starts with migrating Athena from Gandalf to Wuotan, followed by a break of 30 seconds. After that the virtual machine is moved back to Gandalf and the iteration concludes with a break of 30 seconds before the next iteration begins.

The test run of 25 iterations is controlled by a client program that uses the libvirt API to trigger migrations. The migration command returns immediately after live migration finishes allowing us to record the start and the end time as well as the duration of each migration.

We employed two Yokogawa WT210 digital power analyzers to measure the overall AC power consumption of both servers. The devices can measure DC as well as AC power consumption at a rate of 10 Hz and a DC current between 15 μA and 26 A with an accuracy of 0.1 %.

As live migration requires additional resources to perform pre-copy and stop-and-copy rounds, we installed dstat [31] to log CPU, memory and network utilization of the physical as well as the virtual machines.

To obtain the measurements belonging to the same migration, we synchronized the servers time using the Network Time Protocol and use the timestamps of each measurement as links. Figure 4 shows the measurement set for migrating Athena with 1800 MB memory size and 10 MBps network bandwidth.

The power consumption of the source and the destination host rises significantly when the migration starts at 49260 seconds and drops to idle value after the migration finishes at 49450 seconds. The processor's utilization increases in the same fashion, however, as the migration is a an IO-intensive process, the CPU load is very low and does not exceed 10% in both physical machines. In contrast, the source machine's memory usage remains invariable throughout the migration and does not drop until the end. The destination machine's memory usage grows linearly. We explain this behaviour by considering the characteristics of the pre-copy algorithm: Athena occupies 1800 MB main memory and runs continously on the source host during migration. Hence, KVM is not able to free the memory pages on the source machine. In parallel, the hypervisor copies Athena's memory pages to the destination host and causes a linear growth in the memory used on the destination host. After Athena is completly moved and the migration process is finished, KVM can free the memory on the source machine at 49450 seconds, resulting in a memory drop by 1800 MB. The use of 10 MBps network bandwidth on both physical machines can be explained in the same way: KVM transfers memory as fast as the available network bandwidth allows.

V. Evaluation

A. Preliminarily Experiments

To quantify the energy overhead of live migration, we compare the energy consumption of the source and destination host with and without live migration. We measure the power consumption of Wuotan and Gandalf for ten hours continuously while they are idle. In this setting, Wuotan hosts Athena with 2700 MB main memory with no additional co-located virtual machines. Gandalf runs no VM or other applications.

The average overall idle state power consumption is 25.41 W for Wuotan and 25.78 W for Gandalf. It is reasonable to say that both machines consume nearly the same amount of power while idle. The standard deviation for the idle power consumption is 2.32 W for both servers.

B. Measurement Results

To study the influence of VM's RAM size on the energy overhead, we set Athena's main memory to 1 GB, 2 GB and 3 GB. Likewise, we limit the available network bandwidth during migration to 10 MBps and 100 MBps.

To start with, we observe that Athena's size does not much influence the migration duration. Instead, the available network bandwidth does. Migration takes 10 seconds with 10 MBps and 1 second with 100 MBps for all three memory sizes. Looking into dstat's log data, we recognize that independent of the size of VM's entire main memory, only 100 MB data are transferred between the source and the destination host, which matches with the amount of main memory consumed by Athena in an idle state. Hence, we conclude that KVM limits memory transfer in live migration to pages that are actually

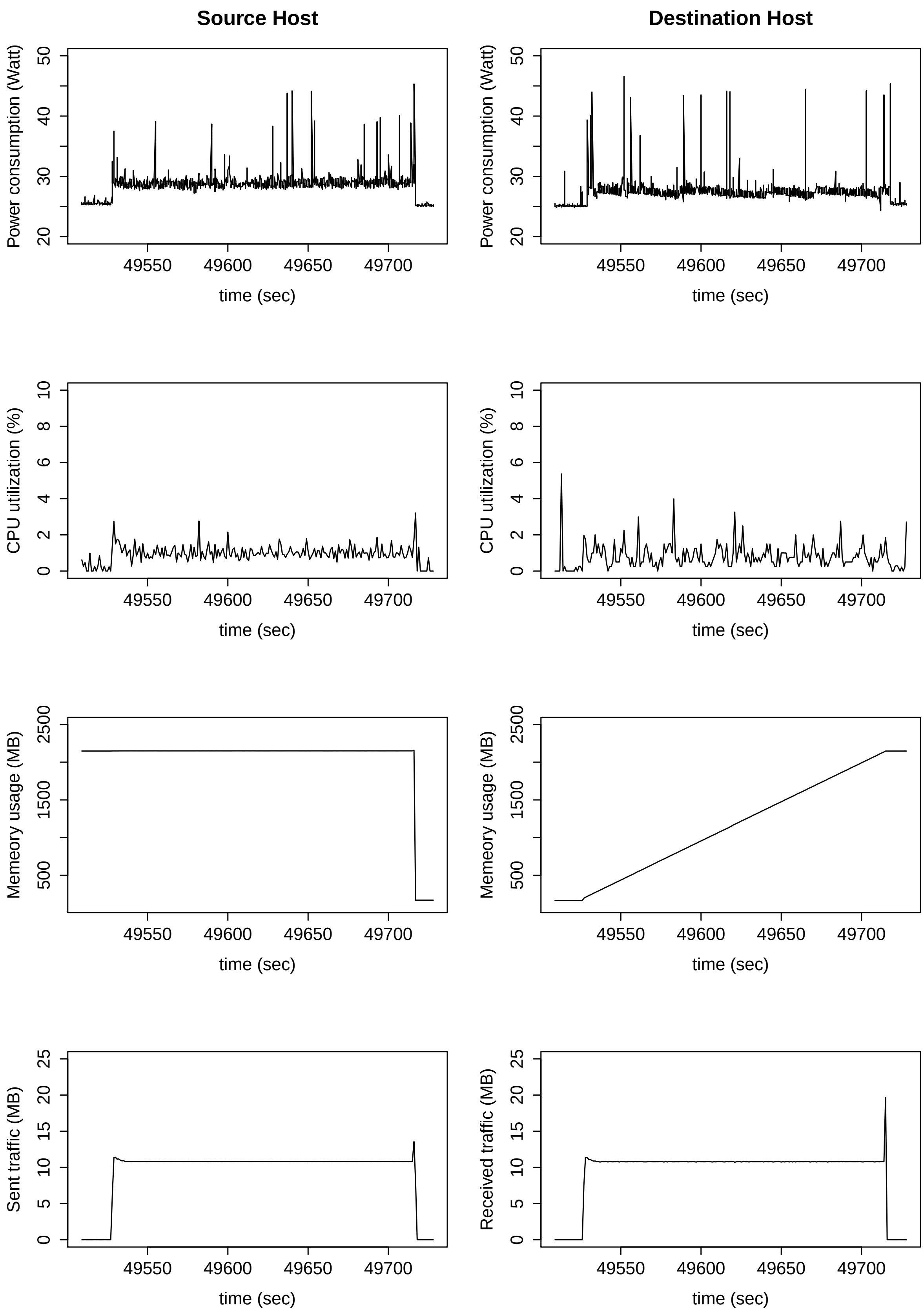

Fig. 4. Power consumption as well as load on processor, main memory and network interface card of source and destination host during the migration of Athena with memory size of 1800 MB and 10 MBps network bandwidth.

in use, i.e. allocated by the VM. Reserved but free memory is not copied.

To occupy Athena's entire reserved memory, we run a memory allocator (see Algorithm 1) we developed. The memory allocator generates Mega Bytes of data to be stored by the VM. To set one Mega Byte main memory, we store 262,144 integers (one integer requires 4 byte in memory, hence 262,144 * 4 byte = 1,048,476 byte = 1 mega byte) with arbitrary values (row 2 - 4). Because swapped memory pages are not moved during live migration, we prevent swapping in row 5 using the function "$mlock$". After allocating the required amount of memory, an infinite loop operation hinders the program from terminating, because termination will re-free all the memory content we allocated.

Remarkably, the operation system does not allow a VM to consume the entire memory its allocated to use. We obtained through experiments that there are upper bounds to the permissible memory size that can be consumed by a VM. These are 800 MB, 1800 MB and 2700 MB for 1 GB, 2 GB and 3 GB main memory, respectively. Hence, in our experiment, the term "VM size"' refers to the size of the actually consumed (utilized) main memory by a VM.

Algorithm 1 Memory Allocator

```
1: for i = 1 → nMegaBytes do
2:     for j = 1 → 262144 do
3:         int *pt ← (int) malloc(sizeof(int))
4:         *pt ← j + i
5:         mlock(pt, sizeof(int))
6:     end for
7: end for
8: while true do
9:     sleep(5)
10: end while
```

1) Power Consumption: Figure 5 shows the arithmetic average of the power consumed by the source and destination hosts during migration. In contrast to the approaches that assume virtual machine live migration is for free, we make the following observation:

1) The cost of migration in terms of power consumption is not negligible and power consumption during migration exceeds the idle power consumption by up to 63%.
2) In all the configurations, the destination server consumes more power than the source server.

To analyze the power consumption in more detail, we employ the probability distribution functions (CDF) of the power consumptions of the two servers during migration. The CDF approach considers the power consumption as a random variable **p**. This is a fitting consideration since it is not possible to give a complete account as to why the power consumption of a server or a component thereof behaves the way it does. The CDF, or simply the distribution function is defined as:

$$F_p(p) = P\{\mathbf{p} \leq p\} \tag{2}$$

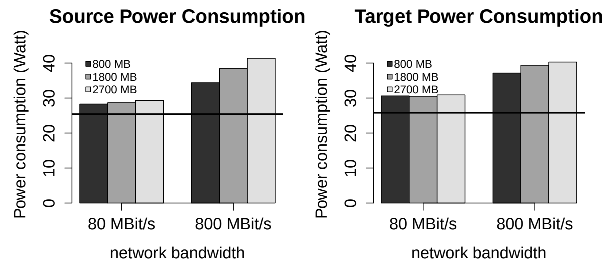

Fig. 5. Average power consumption of source and target host during migration compared with average power consumption in idle mode (horizontal, dark line)

where p is a real number. In other words, the distribution function expresses the probability that the random variable **p** has a value less than or equal to a certain real number p. Because it is a cumulative function, $F_p(p)$ is monotonic increasing, so that for any $\{p_1, p_2 | p_2 > p_1, F_p(p_2) \geq F_p(p_1) \quad \forall p_2, p_1\}$. Moreover, for our case, $F_p(0) = 0$ and $F_p(\infty) = 1$.

Figure 6 displays the CDF of the power consumption of the two physical machines during the live migration of Athena in different configurations. A close examination of these functions reveals the following:

1) The power consumption of both servers during migration is more influenced by the available (utilized) network bandwidth than by the size of the virtual machine and increases with an increment in the available network bandwidth.
2) In general, the VM size contributes little to the power consumption of both servers.

The left part of Figure 6 shows the distribution functions $F_P(p)$ of the power consumption of the source server during the migration of 800, 1800 and 2700 MB virtual machines when the source server was using 10 and 100 MBps network bandwidth. The two figures clearly display the role the network bandwidth played during migration. Regardless of the size of the virtual machine, 80% of the power consumption of the server was below 30 W when the network bandwidth was 10 MBps. Moreover, there was no significant difference in power consumption during the migration of 800 MB, 1800 MB and 2700 MB virtual machines.

The same cannot be said in the case of 100 MBps bandwidth utilization. Firstly, the power consumption of the server increases significantly for all virtual machine sizes, this increment amounting on average to 15 W. Secondly, this time the power consumption of the server is influenced by the size of the virtual machines. For example, the source server is 40% of the time near an idle state (consuming $\approx$ 25 W) when the VM size is 800 MB, whereas it is 20% and 15% of the time in the same state when the VM is 1800 MB and 2700 MB, respectively. Unlike the case when the bandwidth is 10 MBps, the server is almost in two distinct states, either it is active, consuming between 40 and 50 W, or, in near-idle state, consuming around 25 W.

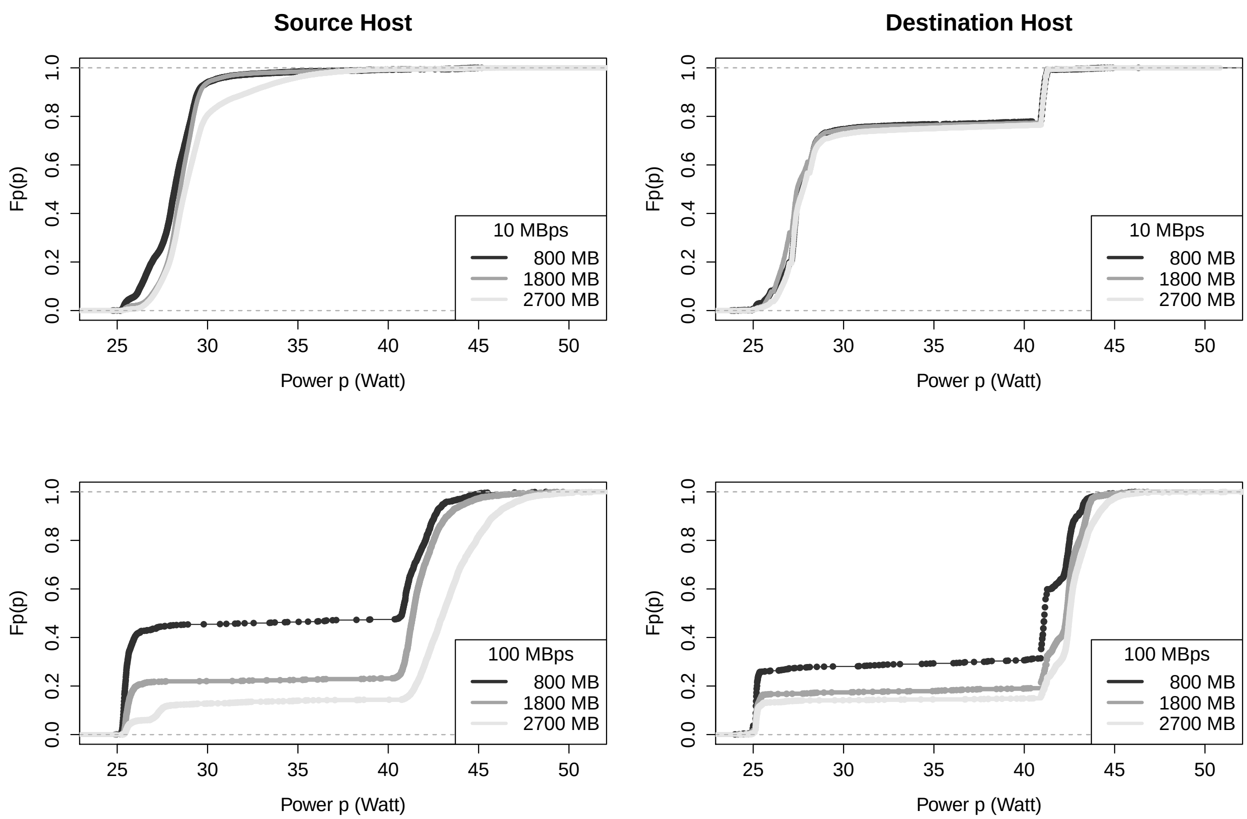

Fig. 6. The distribution function ($F_P(p)$) of the power consumption of the source and destination server during a migration. Top: When the bandwidth was 10 MBps. Bottom: When the network bandwidth was 100 MBps.

Network BW (MBps)	10			100		
VM Size (GB)	1	2	3	1	2	3
$P\{p \leq 30W\} \approx ?$	0.8	0.8	0.8	0.5	0.22	1.5
$P\{p \leq ?W\} \approx 1:0$	30	38	38	45	45	50

TABLE I
SUMMARY OF THE DISTRIBUTION FUNCTIONS OF THE POWER CONSUMPTION OF SOURCE SERVER DURING A LIVE MIGRATION OF A VM.

Network BW	10 MBps			100 MBps		
VM Size (GB)	1	2	3	1	2	3
$P\{p \leq 30W\} \approx ?$	0.78	0.78	0.78	0.3	0.17	1.5
$P\{p \leq ?W\} \approx 1:0$	42	42	42	45	45	45

TABLE II
SUMMARY OF THE DISTRIBUTION FUNCTIONS OF THE POWER CONSUMPTION OF DESTINATION SERVER DURING A LIVE MIGRATION OF A VM.

Table I summarieses the distribution functions of the power consumption of the source server.

The right part of Figure 6 displays the distribution functions of the power consumption of the destination server while it received the 800, 1800 and 2700 MB virtual machines from the source server, utilizing 10 and 100 MBps network bandwidth. Once again, the size of the VM plays no considerable role while the server is utilizing a 10 MBps bandwidth. However, contrary to the previous case, the VM size plays a minor role when the destination servers network utilization is 100 MBps bandwidth. On the other hand, in both configurations (BW = 10 MBps and BW = 100 MBps), the power consumption of the destination server is greater than the power consumption of the source server for the corresponding configurations. Less than 80% of the power consumption is below 30 W when the server is utilising 10 MBps bandwidth. The probability that the power consumption of the destination server is between 30 and 40 W is almost zero. Furthermore, 20% of the time the power consumption of the destination server is above 40 W. Similarly, only 30% of the time the servers power consumption is below 30 W when its network utilisation is 100 MBps. More than 60% of the time, the servers power consumption is above 40 W.

Table II summarizes the distribution functions of the power consumption of the destination server.

2) Migration Time: We measured the migration time when the VM size is 800, 1800 or 2700 MB and when the servers available bandwidth is 10 or 100 MBps. Figure 7 summarizes the arithmetic average of the migration time and make the following observations:

1) The migration time of an idle VM varies linearly with the VM size.
2) The migration time of an idle VM reduces with an

increasing available (utilized) network bandwidth.

Figure 7 shows that the migration time is 90, 190 and 280 seconds when the VM size is 800, 1800 and 2700 MB, respectively, and the network bandwidth is 10 MBps. The average migration time for a VM size of 800, 1800, and 2700 MB with the network bandwidth of 100 MBps is 10, 20 and 30 seconds, respectively. These values clearly demonstrate the role of a VM size and the network bandwidth in live migration of virtual machines with the pre-copy approach. As in idle mode, if almost no page dirtying happens, no pages have to be copied twice and pre-copy is limited to only one iteration to move the VM's entire memory content from the source to the target host. Hence, if a VM size is doubled, for example from 800 to 1800 MB, twice as many bits are copied, resulting in a migration time that is twice as long. In contrast, a higher network bandwidth allows moving more bits within a specific time period, reducing the time required for memory transfer.

3) Energy overhead: We calculate the migration's energy overhead by multiplying the power overhead with migration time, as:

$$E_{mig} = ((P_{s,mig} - P_{s,id}) + (P_{d,mig} - P_{d,id})) * t_{mig} \quad (3)$$

where $P_{s,id}$ and $P_{d,id}$ are the idle power consumption of the source and destination machines, respectively; $P_{s,mig}$ and $P_{d,mig}$ are the migration power consumption of source and destination host, respectively; and t_{mig} is the migration time. Based on this expression, we make the following observations:

1) The cost of live migration in terms of energy is not negligible and varies with the network bandwidth and the VM size.
2) The energy overhead increases with an increasing VM size and reduces with an increasing network bandwidth.
3) In general, the network bandwidth influences the energy overhead more than the size of the VM.

Figure 7 shows the average energy overhead of a live migration as a function of the network bandwidth and the VM size. The energy overhead is 732, 1593, and 2662 Ws for 800, 1800 and 2700 MB VM size, respectively, when the network bandwidth is 10 MBps. For 100 MBps network bandwidth, the figure changes to 203, 539 and 908 Ws for 800, 1800 and 2700 MB VM size, respectively. If the number of dirtied memory pages increases, the migration time takes longer and the energy overhead increases, and the increment slop depends on the VM size. However the energy overhead drops with an increasing network bandwidth, even thought, the power consumption of the servers may be rising. This is because the power overhead does not necessarily increase with increasing network bandwidth at the same rate the migration time drops. Hence, the network bandwidth dominates the energy overhead of a live migration.

VI. Conclusion

In this paper we investigated the energy cost of live migration and analyzed the role of VM size and network bandwidth in the energy consumption of hosting servers. We migrated virtual machines of 800, 1800 and 2700 MB memory sizes when using 10 and 100 MBps network bandwidth. We measured the migration time and the power consumption of the source and the destination hosts and compared these values with the power consumption of the hosts in an idle state. Our experiments' observations are the following:

1) The power consumption of both the source and the destination servers was anything but negligible and was much influenced by the available (utilized) network bandwidth than by the size of the virtual machine. The Power consumption increased with an increment in the utilized bandwidth.
2) The time for migration varied with the VM size and the network bandwidth. Increasing the VM size increased the migration time and increasing the network bandwidth reduced the migration time proportionally.
3) By implication, the energy consumption of live migration of virtual machines should not be ignored. The energy overhead rises with an increment in the VM size and drops with an increment in the available network bandwidth.

In future work we will extend our experiment to analyze the energy consumption of the various subsystems of the source and destination servers in order to better manage when and which VM should be migrated during service consolidation.

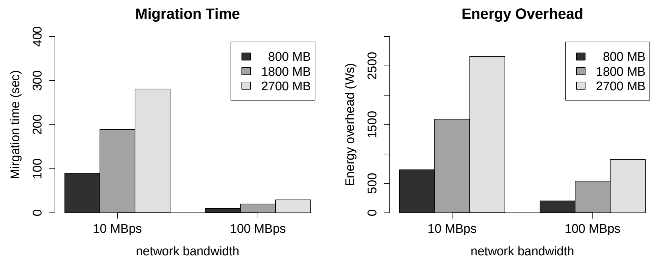

Fig. 7. Migration time (left) energy overhead (right) for migration of 800, 1800, 2700 MB sized virtual machines at 10 and 100 MBps network bandwidth.

Acknowledgment

This work has been partially funded by the European Regional Development Fund (ERDF), the European Social Fund (ESF), and the German Free State of Saxony under project agreement number 080949277

The work has also been partially funded by the German Research Foundation (DFG) under project agreement: SFB 912/1 2011.

References

[1] 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," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 164–177, Oct. 2003. [Online]. Available: http://doi.acm.org/10.1145/1165389.945462

[2] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, ser. NSDI'05. Berkeley, CA, USA: USENIX Association, 2005, pp. 273–286. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251203.1251223

[3] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and gray-box strategies for virtual machine migration," in *Proceedings of the 4th USENIX conference on Networked systems design anbd implementation*, ser. NSDI'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 17–17. [Online]. Available: http://dl.acm.org/citation.cfm?id=1973430.1973447

[4] B. Gerofi, H. Fujita, and Y. Ishikawa, "An efficient process live migration mechanism for load balanced distributed virtual environments," in *Cluster Computing (CLUSTER), 2010 IEEE International Conference on*, sept. 2010, pp. 197 –206.

[5] M. Andreolini, S. Casolari, M. Colajanni, and M. Messori, "Dynamic load management of virtual machines in cloud architectures." in *CloudComp*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, D. R. Avresky, M. Diaz, A. Bode, B. Ciciani, and E. Dekel, Eds., vol. 34. Springer, 2009, pp. 201–214. [Online]. Available: http://dblp.uni-trier.de/db/conf/cloudcomp/cloudcomp2009.html AndreoliniCCM09

[6] H. Mi, H. Wang, G. Yin, Y. Zhou, D. Shi, and L. Yuan, "Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers," in *Services Computing (SCC), 2010 IEEE International Conference on*, july 2010, pp. 514 –521.

[7] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "Enacloud: An energy-saving application live placement approach for cloud computing environments," *2012 IEEE Fifth International Conference on Cloud Computing*, vol. 0, pp. 17–24, 2009.

[8] W. Dargie, "Analysis of the power consumption of a multimedia server under different dvfs policies," *2012 IEEE Fifth International Conference on Cloud Computing*, vol. 0, pp. 779–785, 2012.

[9] W. Dargie, A. Strunk, and A. Schill, "Energy-aware service execution," in *The 36th Annual IEEE Conference on Local Computer Networks*, 2011.

[10] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "Enacloud: An energy-saving application live placement approach for cloud computing environments," in *Proceedings of the 2009 IEEE International Conference on Cloud Computing*, ser. CLOUD '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 17–24. [Online]. Available: http://dx.doi.org/10.1109/CLOUD.2009.72

[11] *Efficient Resource Management for Cloud Computing Environments*. Chicago, IL: IEEE, 08/2010 2010. [Online]. Available: http://grids.ucs.indiana.edu/ptliupages/publications/10-ajy-igcc.pdf

[12] P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem, "Adaptive control of virtualized resources in utility computing environments," in *Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*, ser. EuroSys '07. New York, NY, USA: ACM, 2007, pp. 289–302. [Online]. Available: http://doi.acm.org/10.1145/1272996.1273026

[13] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, and Y. Chen, "Greencloud: a new architecture for green data center," in *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, ser. ICAC-INDST '09. New York, NY, USA: ACM, 2009, pp. 29–38. [Online]. Available: http://doi.acm.org/10.1145/1555312.1555319

[14] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proceedings of the 20th international symposium on High performance distributed computing*, ser. HPDC '11. New York, NY, USA: ACM, 2011, pp. 171–182. [Online]. Available: http://doi.acm.org/10.1145/1996130.1996154

[15] A. Strunk, "Costs of virtual machine live migration: A survey," in *Services (SERVICES), 2012 IEEE Eighth World Congress on*, june 2012, pp. 323 –329.

[16] S. Akoush, R. Sohan, A. Rice, A. Moore, and A. Hopper, "Predicting the performance of virtual machine migration," in *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2010 IEEE International Symposium on*, aug. 2010, pp. 37 –46.

[17] T. Imada, M. Sato, and H. Kimura, "Power and qos performance characteristics of virtualized servers," in *Grid Computing, 2009 10th IEEE/ACM International Conference on*, oct. 2009, pp. 232 –240.

[18] Y. Wu and M. Zhao, "Performance modeling of virtual machine live migration," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, july 2011, pp. 492 –499.

[19] Y. Kuno, K. Nii, and S. Yamaguchi, "A study on performance of processes in migrating virtual machines," in *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*, march 2011, pp. 567 –572.

[20] S. Kikuchi and Y. Matsumoto, "Performance modeling of concurrent live migration operations in cloud computing systems using prism probabilistic model checker," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, july 2011, pp. 49 –56.

[21] A. Verma, G. Kumar, and R. Koller, "The cost of reconfiguration in a cloud," in *Proceedings of the 11th International Middleware Conference Industrial track*, ser. Middleware Industrial Track '10. New York, NY, USA: ACM, 2010, pp. 11–16. [Online]. Available: http://doi.acm.org/10.1145/1891719.1891721

[22] Y. Kuno, K. Nii, and S. Yamaguchi, "A study on performance of processes in migrating virtual machines," in *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*, march 2011, pp. 567 –572.

[23] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *Proceedings of the 1st International Conference on Cloud Computing*, ser. CloudCom '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 254–265.

[24] G. Jung, M. Hiltunen, K. Joshi, R. Schlichting, and C. Pu, "Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures," in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, june 2010, pp. 62 –73.

[25] T. V. T. Duy, Y. Sato, and Y. Inoguchi, "Performance evaluation of a green scheduling algorithm for energy savings in cloud computing," in *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, april 2010, pp. 1 –8.

[26] G. Dhiman, G. Marchetti, and T. Rosing, "vgreen: a system for energy efficient computing in virtualized environments," in *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design*, ser. ISLPED '09. New York, NY, USA: ACM, 2009, pp. 243–248. [Online]. Available: http://doi.acm.org/10.1145/1594233.1594292

[27] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "Enacloud: An energy-saving application live placement approach for cloud computing environments," *2012 IEEE Fifth International Conference on Cloud Computing*, vol. 0, pp. 17–24, 2009.

[28] "Fedora 15. http://fedoraproject.org/."

[29] "Freenas: Storage for open source. http://www.freenas.org/."

[30] "Libvirt: The virtualization api. http://fedoraproject.org/."

[31] "Dstat: Versatile resource statistics tool. http://dag.wieers.com/home-made/dstat/."