Energy-Aware Service Consolidation: Monitoring and Analysis of Parameters

Florian Niedermeier, Gergő Lovász, Hermann de Meer
University of Passau
Innstrasse 43, 94032 Passau, Germany
{florian.niedermeier,gergoe.lovasz,demeer}@uni-passau.de

Stefan Dambeck
:a:k:t: Informationssysteme AG
Dr.-Emil-Brichta-Strasse 7, 94036 Passau, Germany

***Abstract*—While Information and Communication Technology (ICT) has contributed to the reduction of energy consumption in many ways, the energy consumption of ICT itself is rapidly growing, especially in data center environments. This paper presents two important modules of an energy-aware resource management within a virtualized data center: a monitoring and an analyzer module. The monitoring module provides monitoring data from physical and virtual machines. This data is processed by the analyzer module to derive high-level information, which is used to identify situations that require a redistribution of virtual machines. The analyzer module then triggers the virtual to physical mapping process, handled by the optimizer module. The optimizer module calculates near energy-optimal virtual to physical resource mappings, based on service resource requirement profiles and hardware power consumption modeling. The geared operation of these modules enables a tightly consolidated data center operation which does not impair quality of service and leverages heterogeneity in data center environments.**

I. Introduction

Energy consumption of ICT infrastructures has become an increasingly important topic over the last years. The reasons are twofold: On one hand, the high energy consumption is an ecological problem. There is a world wide effort to reduce greenhouse gas emissions. On the other, rising energy prices are putting an increasing economical pressure on companies to turn to green IT products. Today, the acquisition costs of data center servers are lower than the cost introduced by their energy consumption over their lifetime. In the recent past, virtualization has been adopted in a large number of data centers. The primary reason to employ this concept was to enable consolidation. Using consolidation, multiple services running in different virtual machines (VMs) can be hosted on a single server. This way, the number of physical servers required to host a set of services can be reduced, which in turn reduces costs for hardware acquisition. However, when employing virtualization, the new challenge of VM mapping arises: Given a set of services running in VMs, how can these VMs be assigned to physical hosts in an optimal way? This mapping of virtualized services to the physical substrate infrastructure is currently statically and manually assigned. However, as these mapping decisions are static, any physical server has to be able to host its assigned virtual machines even in times of peak load. This limits consolidation density and causes large parts of server resources to be idle during times of low load. Additionally, without detailed knowledge of the resource demand (e.g. CPU time, I/O bandwidth) of services, consolidation is often applied oblivious of the underlying hardware. This leaves headroom for further savings which can only be exploited through the use of an energy-aware resource management that takes into account both the dynamic resource demand of services and the physical hardware characteristics. Such a resource management can be used to achieve a significantly higher consolidation density without sacrificing the experienced service quality.

As service resource demand highly depends on the number of user requests to that service, it can generally not be accurately predicted. Therefore, an important part of such an energy-aware resource management is a monitoring module which constantly supervises the current utilization of hardware resources to avoid over- or underutilization. The monitoring module has to be capable of dealing with heterogeneous server hard- and software and provide monitoring data in a uniform way for further processing.

The monitoring data has to be analyzed and events like formation of hotspots or overloading of servers have to be identified. Additionally, by monitoring VMs hosting certain services, the resource demand of services over time can be profiled to identify patterns. This helps to take more sophisticated and sustainable mapping decisions later.

Conditions like hotspots or overutilization can be mitigated by reevaluating the current mapping of virtual to physical resources. Data centers generally consist of heterogeneous servers, each with distinct hardware characteristics. To achieve higher energy-efficiency, these characteristics have to be accounted for when deriving a mapping of virtual to physical resources.

The remainder of this paper is structured as follows: Section II presents previous research in the area of monitoring and analysis. Section III presents the overall architecture of the envisioned energy-aware resource management. Sections IV and V provide a detailed description of the monitoring and analyzer module respectively. Finally, the conclusion is presented in section VI.

II. Related Work

A. Parameter Monitoring

Network and host monitoring is a widespread method used to assess reliability and performance in data centers.

Consequently, a number of tools for monitoring purposes are available. The Multi Router Traffic Grapher (MRTG) software [1] is a very specialized monitoring solution written in Perl programming language and tailored to router traffic monitoring. For information gathering, MRTG relies on the Simple Network Management Protocol (SNMP) [2], which must also be supported by the end device. By default, it only monitors data input and output of the monitored devices.

Cacti [3] is an open source frontend to RRDtool's [4] data storage and graphing functionality. Cacti polls services at predetermined time intervals and gives a graphical representation of the obtained data. It can be used to monitor metrics like CPU load and network bandwidth utilization. It is also able to communicate via SNMP. The Cacti frontend is PHP-based and multi-user capable. Cacti can use two different backends, on one hand a simple PHP script which is primarily suited for small networks. On the other hand, there is a C-based implementation called ”spine”, which is much more scalable.

A more versatile monitoring approach is realized by the Xymon [5] software which has been released under GNU General Public License. Xymon is a tool for monitoring servers, applications and networks. It can monitor information about computer hardware, the running applications, and network connectivity. The gathered information is visualized on several web pages.

Nagios [6] is an open source host and network monitoring software. It was written and is currently maintained by Ethan Galstad. It is free software, licensed under the terms of the GNU General Public License version 2. As Nagios incorporates a plug-in system, its monitoring capabilities can be largely extended to meet most system specific requirements. Using the Nagios Remote Plug-in Executor (NRPE) [7], it is possible to monitor remote hosts by using scripts hosted on the monitored system. Via NRPE, it is possible to monitor resources such as disk usage or CPU load.

Table I lists the performance characteristics of the described tools. The monitoring module uses Nagios as it supports the two main features needed. On one hand, Nagios can be configured to directly write its monitoring data to a SQL database, which is crucial for later time-series analysis. On the other, Nagios' monitoring capabilities are largely extendable via a plug-in system. This way, system-specific monitoring can be realized.

B. *Parameter Analysis*

Using monitoring data to trigger power management actions has been intensively researched. Wood et al. [8] used VM migration to mitigate hotspots in virtualized environments. They compared a black-box and gray-box monitoring, the former relies on monitoring data gathered from the virtual machine monitor (VMM) only, whereas the latter also incorporates VM level monitoring data. They characterize virtual and physical machines by creating distribution and time series profiles. The system triggers management actions based on performance parameters (e.g. CPU utilization). However, it implies fixed thresholds for migration triggering. Rodero et al. [10] use dynamic voltage and frequency scaling (DVFS), processor pinning and migration techniques to mitigate hotspots and increase energy-efficiency. Ramos et al. [11] use on-line temperature and performance prediction to orchestrate load redistribution and DVFS in servers. Both of these approaches trigger management actions based on temperature thresholds. However, they do not incorporate service performance as a trigger for management actions. Singh et al. [9] present an algorithm called VectorDot for load balancing based on host monitoring. It considers multiple server and network resources as VM migration triggers, however they do not consider dynamic shutdown or startup of physical machines.

III. Energy-Aware Consolidation Approach

Many data centers are already employing virtualization to reduce hardware costs by consolidating services on a smaller number of servers. Apart from consolidation, virtualization is a key component in providing a more flexible, energy-efficient mapping of virtualized services onto a physical substrate infrastructure. The primary gain in flexibility is due to the possibility of online virtual machine migration, which enables a service to be relocated to an arbitrary physical node inside the same subnet without service interruption. Depending on the overall load of the data center, services can be consolidated at different densities. In times of low or average overall load, a low number of physical servers can host the virtualized services, while other parts of the physical infrastructure can be shut down. This is due to the common practice of data center overprovisioning, meaning hardware resources are designed for peak load operation and include a further safety margin. Apart from the virtualization approach that is required to enable the seamless migration and consolidation of services within the physical substrate infrastructure, the suggested energy-aware resource management consists of three building blocks:

1) **Monitoring module:** The monitoring module is necessary to supervise the energy-relevant parameters of the physical substrate infrastructure as well as the virtualized services inside the data center. It is capable of monitoring heterogeneous servers, in terms of hard- and software. Also, the monitoring data is provided in a uniform way for further processing. The monitoring frequency has to be adjustable to adapt to the specific demands of different data center environments.
2) **Analyzer module:** The analyzer module interprets the raw monitoring data to deduce necessary management actions. The analyzer has the responsibility to ensure that service consolidation is performance conserving, meaning the perceived quality of service is not degraded. This is achieved via dynamic resource utilization thresholds, which trigger a migration of services. Additionally, the analyzer stores relevant monitoring information and state changes within a dedicated data base. This data is used to build profiles of resource usage and dynamic

device characteristics that are based on historical and current states of the virtualized services and the ICT infrastructure.

3) **Optimizer module:** In this paper the optimizer is regarded as a black box, details on the implementation are out of scope. The optimizer component calculates near energy-optimal mappings of virtualized services to physical resources that do not violate the resource requirements of the services and therefore do not affect performance. It estimates the energy consumption of various possible mappings and chooses a nearly optimal solution. For the calculation of such a near-optimal mapping, the optimizer uses power consumption models of the physical hardware and service requirement models of the virtualized services.

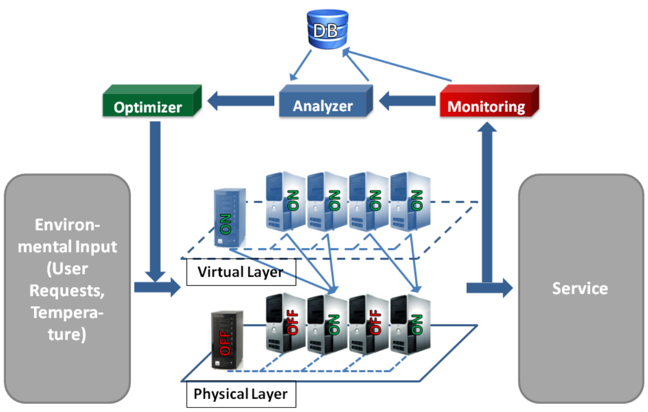

Fig. 1. Energy management feedback loop

All of the described components are illustrated in Figure 1. The physical layer represents the physical hardware of the infrastructure without virtualization and indicates the different energy-states of the server hardware. The virtual layer plays the role of a virtual counterpart of the physical infrastructure. The users of the infrastructure interact only with virtualized nodes and are not aware of the virtualization or the presence of an underlying physical infrastructure layer. The energy-aware management interacts with both, the physical and the virtual layer.

IV. The Monitoring Module

A. Architecture

The monitoring module is the main communication link between the energy-aware resource management and both the virtual and physical layers. Multiple commonly used monitoring tools are evaluated regarding their applicability for an energy-oriented monitoring system in Table I.

	MRTG	Cacti	Xymon	Nagios
SNMP Support	Yes	Yes	Plug-in	Plug-in
Client Specific Agents	No	No	Xymon Agent	NRPE Agent
MySQL Support	No	Yes	No	Yes
Network Service Check	No	No	Yes	Yes
Web Interface	Not included	Yes	Yes	Yes
Web-based Configuration	No	Yes	No	Plug-in
Graphical Monitoring	Yes	Yes	Plug-in	Nagios Grapher
Threshold Alerting	Yes	Plug-in	Yes	Yes
Threshold-based Actions	Yes	No	Yes	Yes

TABLE I
MONITORING TOOL PROPERTIES

Out of the nine investigated features, three are especially crucial for an energy-aware resource management.

- **SNMP Support:** To enable low overhead communication with e.g. integrated service processors (SPs) of modern data center servers, the monitoring software has to provide a method for SNMP communication.
- **Client Specific Agents:** As the monitoring module is designed to handle heterogeneous environments, at least some monitoring can not be implemented in a uniform way. E.g. performance statistics have to be obtained by operating system specific commands. So, client specific agents are necessary to provide a generic interface for the monitoring module.
- **MySQL Support:** For further analysis of the collected monitoring data the use of a relational database system is reasonable. Especially for the creation of service profile based on historic resource utilization data, the possibility to store monitoring values in a database is indispensable. Also, it simplifies and accelerates the access to the data and makes it easily accessible over the network.

Monitoring tools like Nagios build upon a solid client-server architecture and include a proven communication subsystem. But when it comes to sophisticated, automatic actions based on the monitored data, the integrated mechanisms do not satisfy the demands. Simple definable threshold-driven actions like the base features of Nagios are not enough to implement a resource management system for virtual environments.
Nagios builds the basis for the monitoring module, mostly because of the existing database integration and the broad use of this tool, which is also visible by the wide range of freely available additional extensions and plug-ins. However, the monitoring data analysis is performed in a separate module.

B. Parameters

Even basic performance values deliver valuable information on resource usage and build the basis for management decisions and actions of the analyzer and optimizer modules. Monitoring information can be gathered at two levels: Physical machines and VMs. Physical machine monitoring is necessary to evaluate the state of the hardware substrate infrastructure, the following list of monitoring parameters names some crucial values which have to be considered:

- CPU load: A machines' CPU load is important in two aspects. It is one of the key performance indicators, and therefore has to be monitored to be able to give performance estimates. Additionally, it is the primary contributor to the dynamic power consumption of a server.
- RAM usage: RAM usage has to be monitored, as it contributes both to power consumption and is a limiting

factor for virtual machine consolidation.

- Storage I/O: As storage medium power consumption depends largely on its utilization, monitoring is necessary. It also limits the consolidation of I/O bound services.
- CPU temperature: The processor temperature is an important indicator to detect hotspots.
- Fan speed: High speed fans are a significant contributor to overall power consumption. Also, when evaluated together with CPU temperature, it is a good indicator for the data center ambient temperature.

When monitoring virtual machines, less parameters have to be considered. Virtual machine monitoring is primarily done to create a service profile, e.g. record its resource consumption.

- CPU load: Apart from the average CPU load, the peaks have to be recorded in order to evaluate CPU load stability. Also, as most servers have a multi-core architecture, it is necessary to know if a service is capable of exploiting multi-threaded execution.
- RAM Usage: To decide which physical host is able to accommodate the virtual machine, RAM usage has to be monitored.
- Storage I/O: Especially in non-shared storage environments, the utilization of the storage medium can have a significant impact on virtual machine placement. E.g. in a high utilization case, placement on a physical machine employing solid state discs is feasible.

When looking at the Nagios software, it can provide the necessary monitoring data by using plug-ins. Some monitoring data can only be gathered from inside a VM or on VMM level, this can be enabled by using Nagios Remote Plug-in Execution (NRPE) or a similar plug-in. The consistent use of the NRPE component both on the physical (VMM) and the virtual (VM) layer leads to standardized performance data for all elements of the infrastructure. Some additional monitoring data (e.g. fan speeds on certain mainboards) can only be collected from the physical host via a service processor (SP). This data can be gathered via the SNMP plug-in.

V. The Analyzer Module

A. Optimizer Triggering

The analyzer module observes and interprets the current mapping of virtual to physical resources as well as environmental monitoring data. This is necessary as the resource demand of virtualized services is in general dynamic and dependent on the load of the service. Therefore, multiple situations may arise which can lead to a need of VM redistribution. Also, environmental conditions like the data center temperature may be fluctuating, leading to conditions that may e.g. require a physical machine to be shut down. The analyzer acts as a triggering module, more specifically it decides when the optimizer module is asked to reevaluate the virtual to physical layer mapping. There are multiple reasons which can cause a trigger event:

1) Regular reevaluation: The regular reevaluation makes sure that virtual machine consolidation is kept at a high level while maintaining the required performance. More specifically, if physical machines remain at a stable medium load for a long time and do not trigger any conditions listed below, consolidation possibilities could be wasted. To avoid this situation, the optimizer is called for reevaluation at a low constant rate.

2) Out-of-order reevaluation:

1) **Physical host overutilization:** Many services provided by data centers have a highly dynamic load signature. Especially those applications directly serving a potentially high number of users can cause load patterns that are highly autocorrelated in certain time intervals, e.g. day - night cycles. This non constant resource demand leads to different consolidation possibilities depending on the utilization of the virtualized services. However, as service resource consumption is generally not accurately predictable, regular monitoring is necessary. If a physical host is overloaded, the optimizer module is notified to ease the load on this system. To derive the most feasible monitoring frequency, several aspects have to be taken into account. A lower bound for the monitoring frequency can be derived when comparing monitoring frequency to VM migration time. As overload situations are mitigated by migrating workload to a different physical machine, the time between two monitoring cycles has to be adapted to the migration times of the running VMs. This migration time depends on a number of VM and infrastructure properties, e.g. VM memory size, VM workload, network load or usage of shared storage. Liu et al. [15] showed that a VM with a low memory size of 128 MB still takes an average of more than 7 seconds to migrate. Considering that memory sizes of VMs are usually significantly higher, a migration time of 20 seconds will be assumed. To enable a swift reaction of the optimizer module to an overload situation, we suggest that the monitoring frequency of physical machine load should not exceed 10% of the migration time. However, as the network traffic generated by monitoring itself should not have a significant impact on network performance, an upper limit to monitoring frequency has to be set. As load monitoring tools like mpstat [13] have a maximum resolution of one measurement per second, this is also a reasonable limit for the overall monitoring frequency. In the assumed case, this would result in a suggested monitoring frequency of 0.5 - 1 Hz.
   A lower frequency would lead to an increased delay between the occurrence and the detection of an overload situation. This results in an increased time in which at least one of the physical servers resources is in an overutilized state. This in turn could result in a degraded QoS, e.g. increased delay or jitter, which could endanger the fulfillment of service level agreements (SLAs).
   However, choosing a monitoring frequency exceeding the recommended range would lead to heavy network

traffic and rising resource consumption by the resource management itself. As some protocols used for monitoring, e.g. SNMP, use UDP as transport protocol, this could not only affect network performance but also impair the monitoring itself due to packet loss.

2) **Physical host underutilization:** Similarly to the overutilization case, a physical host can become underutilized due to workload dynamics. Although this is not a situation requiring immediate attention (as service performance is not impaired), energy-efficiency can most likely be increased by a higher consolidation density. As underutilization can be detected by evaluating the same parameters as in the overutilization case, the monitoring frequency is already set and does not need to be adjusted, as overutilization is a much more critical state and therefore sets the lower bound to monitoring frequency.

3) **Hotspot detection:** Another factor that may threaten SLA requirements as well as energy-efficient operation is the formation of hotspots. These hotspots may have different causes. Although servers are built to handle a fully loaded operation, there are several reasons that can cause a system to overheat. This should be avoided at all costs, as it causes both performance and energy waste. Performance-wise, overheating can cause both system instability and thermal throttling, leading to either service failure or QoS impairment. Energy-wise, an overheated system will cause fans to speed up both inside of the server and maybe even inside data center air conditioning. So, a system reported to be in a high temperature state has to trigger an optimizer call. By migrating load away from the machine, hotspots can be avoided, increasing both energy-efficiency and reliability. Hotspot detection does not impose new bounds on the monitoring frequency, as temperature is a much slower changing value than e.g. CPU utilization.

B. Profiling

The analyzer employs a database to keep historical data for the development of application profiles. These profiles contain information about the resource usage behavior over time. This includes CPU, network, memory and I/O information. To help the optimizer make more sustainable decisions and to avoid unnecessary migrations, a service is not only described by its average resource demands, but also by the respective standard deviation. This helps to identify the service as stable or volatile in terms of resource demand. Depending on this classification, the thresholds for over- or underutilization are set. Figure 2 shows the comparison of a - in terms of resource demand - volatile and stable service. The volatile service shows strong variance in terms of resource demand, which complicates consolidation choices. When setting an upper bound for resource utilization of a volatile service, this variance has to be considered to avoid hasty migration decisions. Short term threshold violations should be tolerated, however such services have to be provided with additional headroom to avoid service impairment during short term load spikes. In contrast, stable

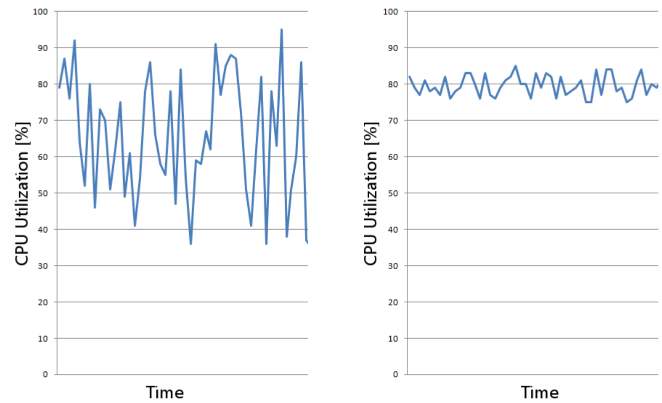

Fig. 2. Comparison of a volatile and a stable service

services allow a higher degree of consolidation density, as their resource demand can be predicted with high accuracy. While a machine hosting primarily stable services can have high thresholds for resource usage before a VM has to be migrated, machines with volatile services cannot be as densely consolidated.

C. Granularity

When looking at the monitoring data flow, four different data streams can be identified. First, raw monitoring data is fed to the analyzer module as well as the database by the monitoring module. The analyzer then extends the information stored in the database by adding high-level information obtained by processing the monitoring data. The optimizer is contacted by the analyzer module in two different situations: First, a reevaluation takes place in regular intervals, depicted as α_1 in 3. Second, when certain conditions are met, the analyzer module triggers an out-of-order message to the optimizer (α_2 in 3). The data streams have different requirements in terms of

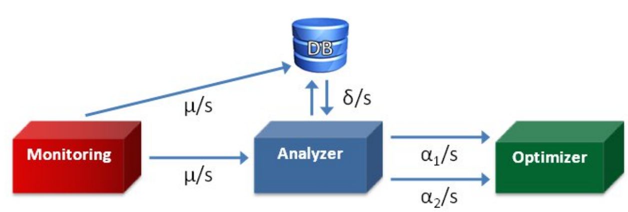

Fig. 3. Data flow inside the energy manager

update frequency. The polling rates must be carefully balanced, as this decision has significant impact. The basic monitoring process has to be repeated fairly fast to make sure that any performance impairing situation is detected in a timely manner. The communication between analyzer and database can be significantly less frequent, as it is only needed to update the dynamic resource usage thresholds. The regular reevaluation α_1 optimizes consolidation in non critical situations, therefore its frequency can be many times lower than the monitoring frequency. Real-time requests - α_2 - do not have a fixed rate, however to avoid network congestion an upper bound is set at 1 Hz. As long as a situation persists that requires immediate action, the message will be repeatedly passed to the optimizer.

VI. Conclusion

To counter the rapid increase in data center energy demand, energy efficiency is a primary concern. This paper introduced the monitoring and analyzer modules of an energy-aware resource management relying on monitoring data. In contrast to previous approaches, it is focused on increasing energy efficiency via dynamic, performance conserving virtual machine mapping. This mapping relies on service profiles and power consumption models. A detailed analysis of existing monitoring tools showed that the Nagios software is best suited to serve as basis for our monitoring module. By extending it, a broad basis of sensor data can be gathered, which serves as foundation for the decisions made in the analyzer module. The analyzer module is used to derive high-level information from the raw monitoring data and trigger a reevaluation of the current virtual to resource layer mapping if the need arises. Future research will include a prototype implementation and evaluation of the envisioned resource management on a Future Internet testbed (e.g. G-Lab [14]).

Acknowledgment

The research leading to these results has received funding from the German Federal Government BMBF in the context of the G-Lab_Ener-G project and from the EC's FP7 framework programme in the context of the EuroNF Network of Excellence (grant agreement no. 216366).

References

[1] MRTG - Tobi Oetiker's MRTG - The Multi Router Traffic Grapher. http://oss.oetiker.ch/mrtg/

[2] RFC3410. Case, J.; Mundy, R.; Partain, D. & Stewart, B., Introduction and Applicability Statements for Internet-Standard Management Framework, IETF, 2002

[3] Cacti: The Complete RRDTool-based Graphing Solution. http://www.cacti.net/

[4] RRDtool - About RRDtool. http://www.mrtg.org/rrdtool/

[5] green : Xymon. http://www.xymon.com/

[6] Nagios - The Industry Standard in IT Infrastructure Monitoring. http://www.nagios.org/

[7] Galstad, Ethan. Nagios: Official NRPE Documentation. http://nagios.sourceforge.net/docs/nrpe/NRPE.pdf

[8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and Gray-box Strategies for Virtual Machine Migration", Proc. of the 4th USENIX Symposium on Networked Systems Design and Implementation, pp. 229-242, 2007

[9] Aameek Singh and Madhukar R. Korupolu and Dushmanta Mohapatra, "Server-storage virtualization: integration and load balancing in data centers", IEEE/ACM, 2008.

[10] Ivan Rodero, Eun Kyung Lee, Dario Pompili, Marc Gamell, Renato J. Figueiredo, Towards Energy-Efficient Reactive Thermal Management in Instrumented Datacenters, 11th IEEE/ACM International Conference on Grid Computing, IEEE, 2010

[11] Ramos, L.; Bianchini, R.; "C-Oracle: Predictive thermal management for data centers", High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on , vol., no., pp.111-122, 16-20 Feb. 2008

[12] Kühn, Paul J., "Systematic Classification of Self-Adapting Algorithms for Power-Saving Operation Modes of ICT Systems", e-Energy 2011 Second International Conference on Energy-Efficient Computing and Networking, 2011

[13] Sebastien Godard, SYSSTAT. http://sebastien.godard.pagesperso-orange.fr/

[14] German Lab. http://www.german-lab.de/

[15] Pengcheng Liu, Ziye Yang, Xiang Song, Yixun Zhou, Haibo Chen, and Binyu Zang, "Heterogeneous Live Migration of Virtual Machines".