Cloud and Grid Part II: Virtualized Resource Balancing

Anna Melekhova¹* and Vladimir Vinnikov²

¹Parallels, Moscow, Russian Federation
²Acronis, Moscow, Russian Federation

Abstract

This second part of the paper is concerned with types and properties of virtual resources in the generalized grid and cloud architectures. We study the contemporary approaches to efficient management of virtual resources like CPU, storage and network along with balancing of power consumption. Our considerations also aim at the performance and security complications arising due to virtualization as well as the means to solve respective problems.

Keywords: Ballooning, Cloud Computing, Cloud Storage, Data Grid, Grid Computing, Grid-Cloud Integration, Hypervisor, Intercloud, Nested Virtualization, Performance Isolation, Supercloud, Virtual Machine, Virtual Network

1. Introduction

The first part of this study was concerned with the generalized models of grid and cloud architectures as well as their mutual convergence to unified grid-cloud architecture. We considered the examples of contemporary grids and clouds and promising mechanisms to achieve their unification into global confederation of clouds, as well as remaining obstacles, bottlenecking the performance of these virtualization mechanisms and the ways to resolve such problems. Therefore the study showed that large and global scale clouds essentially would be comprised of lesser clouds via means of nested virtualization technology⁴ resulting in a recursive configuration pattern. This structure, known as intercloud⁵ or supercloud⁶ would manifest new special aspect such as the ultimate explicit delocalization of computational and storage resources. The study⁷ provides comprehensive survey of the virtual resource management and classify its functional elements, namely: global scheduling of virtualized resources; resource demand profiling; resource utilization estimation; resource pricing and profit maximization; local scheduling of cloud resources; application scaling and provisioning; workload management; and, cloud management systems. However, full virtualization per se does not imply optimal allocation and distribution of requested resources, leading to nonuniformity of their quality. Thus, the cloud design should provide effective resource balancing on multiple abstraction layers. The balance is achievable via resource scavenging, but one has to estimate machine workload first. It appears to be one of the major challenges, since the delocalization complicates traceback to virtual machines at lower levels of abstraction as well as direct access to the underlying physical hardware.

2. Computational Resources for the End-Consumer

Computational resource dates back to machines of early design without permanent storage capabilities. These computers generally were built around CPU and volatile memory modules. During the evolution of

* Author for correspondence
circuit technologies, two antagonistic processes defined hardware architecture. One earlier design paradigm aimed to transfer all specialized operations (e.g. IO, audio and video processing) into separate independent auxiliary units, leaving CPUs only for generic computations. The other more modern paradigm sought to unite all computational capabilities within a single chip or by the means of wide bridge, providing benefits from maximum possible bandwidth.

The computational resource is usually classified into various subcategories. The overall processing performance can be measured as the number of instructions executed within a specified timeframe (millions instructions per second, MIPS). However, this metrics has significant drawback, since not all CPU instructions take equal number of clock cycles. Even same instructions can have different durations on different CPU architectures. Moreover, not all operations from the respective instruction set have computational meaning. More specific arithmetical performance is usually estimated on the number of floating point operations per second (FLOPS). This estimation depends on word size of operands. Complex computations also depend on memory and cache bandwidth, which in case of congestion causes CPU to run idle. For this reason fast volatile memory is designated to computational resource and not to storage one.

The pure arithmetic performance is better suited for relatively large singular computational tasks consuming non-negligible amount of resources. If main computational load is comprised of multiple small jobs, an overall performance is defined more by task distribution ensuring minima of queues and idle cycling across all processing threads. In this case, the performance can be appropriately measured as a number of processed tasks/data per unit of time. This parameter is known as throughput metrics. For large network systems like clusters, grids or clouds the throughput metrics also indicates the scalability. If the throughput metrics exhibits near linear dependence against the number of CPUs, then the scalability is confirmed. In other words, the doubling the nodes leading to near doubling of throughput signifies good scalability potential.

The users qualitatively perceive computational resource as a timeframe between two consequent interactions with supplemented task, i.e. the response delay during update of large spreadsheets, laggy behavior of real-time simulations, or even an application hang-up in an extreme case. The clients of the conventional clusters and grids statically acquire the resource roughly grained in multiple of physical CPUs, CPU-cores or CPU-threads depending on the deployed middleware. However, since the physical machines constituting the grid are heterogeneous, the granularity also can be non-uniform. The distinctive feature of the grids is that the users mainly order CPU-time in bulk to handle large queued non-interactive jobs. In contrast, typical clouds do not explicitly offer the computational resource to users. Instead, the users receive interactive elastic environment adjusting itself to varying client load. The hypervisor provides automated dynamic allocation of the resource and its fine granulation in fractions of CPU, while hiding details from the scope of end-user. The pricing for cloud services is based on actually consumed resources.

The current virtualization capabilities are still insufficient to provide uniform processing power in heterogeneous environments. The work addresses this problem and proposes an approach to standardize the measurement of the processing power by introducing Processing Units. The challenge of implementing nested virtualization with hierarchy of hypervisors in real-time systems of x86 architectures is considered in study.

3. Storage Resource for the End-Consumer

First effective storages were based on magnetic tape devices scaled from a single reader up to corporate data libraries. Up to this day, information retention is one of the computer core functions, provided by the data storage. Storage resource differs significantly from its computational and network counterparts. While the transient and run-time behaviors are common for all resources, the storage virtualization exhibits unique long-term persistence, even if an application is shut down.

The storage resource is characterized by following parameters: storage capacity, IO performance and reliability. The storage capacity is usually self-explanatory, however the virtual storage is often considered unlimited. While for majority of the clients such assumption is valid, the specialized big data cases can exhaust even a petabyte capacity.

The IO performance is commonly measured as data access and transfer rates at various operational modes (e.g. seek rate, sustained read/write rate, etc.). This parameter
for virtual storages relates closely to network resource, much like the computational throughput does, since the managed data have to travel additionally between the client and all processing locations.

The reliability is widely considered the most critical aspect for any storage. Unlike raw capacity, this parameter is strongly statistical and is not additive. Moreover, this metrics manifests inverse dependence on scale. In other words, an increase in storage devices decreases an overall storage health. To overcome this law of statistics the distributed storage has to be provided with special hardware and software solutions implementing various redundancy schemes like RAID, replication or Reed-Solomon coding.

The grids are rarely dedicated solely for data storing, so their clients receive the storage resource as an implicit auxiliary asset for holding of initial data, intermediate data slices at checkpoints, and the results of the job. Depending on the grid operational policies, the storing of arbitrary data can be strictly limited. In contrast, the cloud architectures provide high reliability for storing data. By other words, generally the cloud users purchase the storage capacity, but not the computational power.

4. Network Resource for the End-Consumer

While computational resource is responsible for data transformation, and storage resource answers for a data preservation, then network resource provides not only a data mobility, but also the overall coupling of the distributed system. Without a true connectivity all three clusters, grids and clouds are restricted to their virtualized form on a singular physical machine. Therefore, the network structure significantly affect both computational and storage performances.

The typical metrics for network resource are requests throughput and maximum bandwidth. Throughput is derived from the indicators of the computational counterpart and denotes the number of requests processed within a unit of time. Maximum bandwidth shows maximum amount of data a network is able to convey. First metric shows the resulting operational performance of the constructed system from its input to the output, and the second one measures the quality of the pure transport solutions, including noise protection, optimal topology and so on. These metrics can be used to calculate normalized system efficiency as a ratio of requests throughput to the combination of bandwidth, CPU throughput and IO-performance. The obtained value estimates the fraction of overheads and combined bottlenecks, including coupling expenses for data transformations.

Network virtualization becomes one of the inherent components of future Internet. The virtual Network acts as a primary entity consisting of active virtual nodes and passive virtual links. These two types of elements constitute virtual topology. The mapping of virtual topology onto its physical counterpart referred as substrate network constitutes the Virtual Network Embedding (VNE) problem. The study presents a survey of state-of-art VNE research and provides a taxonomy of current approaches to network virtualization.

From the client point of view, the network resource primarily manifests itself as a communication latency, i.e. the timeframe between sending the request and receiving confirmation of the acceptance. The typical end-users rarely experience explicit management of network resource in distributed systems. The exceptions include specialized distributed applications designed to gather and store data in the areas with thin communication channels (e.g. the gathering of astronomical, meteorological or seismic data from sensor arrays traffic rerouting between cellular stations; arrangement of interacting smart home elements).

Eventually, the client can only notice that Internet Service Providers (ISPs) change their business model to comply the IaaS paradigm of network virtualization. Thus, the ISP splits into the Infrastructure Provider (InP), deploying and maintaining the equipment and the Service Provider (SP), deploying network protocols and offering end-to-end services. The SP itself retains only business role offering customized services, and delegate management roles to the Virtual Network Provider (VNP) aggregating virtual resources from various InPs and the Virtual Network Operator (VNO) operating VN for the SP.

The network resource in its extended interpretation also includes data querying. One of the challenging problems for both grid and cloud systems is an implementation of large scaled distributed databases. The ubiquitous robust relational databases are designed to uphold ACID properties. However distributed transactions across a distributed database result in performance bottlenecks,
since the CAP-theorem burdens ACID capabilities with two-phase commit protocol and either two-phase locking or multiversion concurrency control. The widely considered way to improve performance of data structure resource is to reshape it to a non-relational form\textsuperscript{36} manifested as NoSQL technology. This approach to data management allows the efficient handling of so called Big Data. Big Data usually comes unstructured, and the structuring is unfeasible.

5. Continuity of Operation Principles

By definition, the converged grid-cloud integration maintains continuity of both grid- and cloud-specific services. At the top layer of abstraction an IaaS is represented as a pool of virtual computational, storage and network resources, arbitrary allocable by clients. Ultimately, this interface of grid-cloud integration is undistinguishable for guest OS to that of a single physical machine. Operation of PaaS also resembles OS running on a physical computer in terms of purposes and functioning of utility software. For example, storage integrity checking has been already scaled up from individual HDD scans to data scrubbing across the multitude of nodes. The same functional similarities can be established for CPU loading and memory consumption management. Once PaaS is deployed, the consequent implementation of SaaS is straightforward. Thus, the complete resources virtualization ensures continuity of operation at infrastructural, platform and service application levels\textsuperscript{23}.

6. Generalized Load Estimation

The efficient PaaS operation generally depends on uninterruptible interaction of virtual resources and their fast adjustment to workload changes\textsuperscript{44}. One of PaaS elements achieving this is a resource monitor providing load estimation and optimal balancing to avoid resource congestion. Since PaaS operates on top of IaaS\textsuperscript{32,46} virtual resources should be measured on multiple abstraction layers down to the level of physical computers and networks. The acquisition of the hardware metrics could involve interface transformations at each level due to multiple consequential decouplings\textsuperscript{33}.

The problem of resource congestion originates from the disparity in availability and running costs of various virtual resources. The commodity-grade storage is considered as inexpensive and allocable for long-term usage in any reasonable volumes. Therefore, it is quite unlikely to encounter an unexpected virtual storage starvation. In contrast, burst patterns of guest workloads at their peaks frequently exhaust the CPU and memory leading to considerable performance loss. Since the time-averaged utilization of CPU and memory is not high in usual workloads, the total amount of these nominally granted resources usually exceeds the capacity of underlying real hardware. This technique of excessive resource yield is called overcommit or oversubscription. Such resource distribution among multiple guests is feasible, if simultaneous peak loads are assumed unlikely to occur. In such case, the problem reduces to dynamic resource redistribution, when the fractions of allocated CPU and memory are reclaimed from one guest and committed to another. The paper\textsuperscript{15} presents two feedback controllers dynamically adjusting the CPU allocations of virtualized multi-tier applications and retaining free resources for further demand, thus allowing the achievement of high resource utilization per physical machine.

Since almost every CPU model provides standard instruction for switching itself into a standby state, a virtualization manager reinterprets its occurrences within guest command queue as a resource release and an ability to divert computational power to requesting guest. Contrary to this, there is no equivalent command in a common hardware to mark unused memory\textsuperscript{25}. Thus, a hypervisor is unable to directly measure and manage real memory consumption via hardware and virtualization means. This problem for single tiered virtualization can be solved using classic ballooning technique mentioned earlier, which implements a special driver for implicit interaction between a hypervisor and a guest OS via virtual memory displacement.

The generalized virtualization involves VM nesting of arbitrary depth D, where a hosted hypervisor runs as an application on OS, which, in turn, is deployed on VM governed by a hypervisor of a previous tier. In this case the host physical memory at tier k is allocated from guest virtual memory at tier (k - 1), and the tier-1 host memory corresponds to a physical machine. If memory distribution at each tier relies on ballooning technique then the global load estimation of underlying hardware from the topmost layer would require sequential ballooning chain of length D. The complexity of this generalized problem depends
on D. Classic ballooning implementation for longer chains would result in accumulated inaccuracy and operational lag of high tier memory estimations. These setbacks can be resolved via fast propagation of local-tiered estimations and internal OS counters (e.g. Physical Total, Commit Total, Balloon Size, etc.) from the top hypervisor to the lower one down to the first tier. This fast propagation relies on the improved ballooning technique, considered by $^{25}$. In the paper, the authors propose balloon driver modification to read statistically relevant internal OS counters and deliver them to a hypervisor. The generalized scheme can be described as follows:

1. The tier (k - 1) balloon driver exchanges measurements with tier-k hosted hypervisor via standard means of OS (e.g. pipeline).
2. The tier (k - 1) balloon driver exchanges measurements with tier (k - 1) hypervisor via its interface.
3. Items 1—2 are repeated with k decrementing from D, until (k - 1) reaches the lowest tier considered (e.g. k = 2).

This algorithm generalizes ideas of Melekhova et al (2015), and allows a hypervisor at arbitrary tier to receive internal OS counters, accumulated across any and all tiers of interest. One can feed this data to the decision-making control script to build a prediction upon an autoregressive model and/or a neural network.

7. Generalized Load Balancing

The applied problem of load balancing can be divided into following challenges $^5$

• Prediction of Virtual Environment Configurations and Operating Schedule;
• Behavior Prediction of Application Services;
• Flexible Mapping of Services to Resources;
• Economic Models Driven Optimization Techniques;
• Scalable Monitoring of System Components.

A set of virtual environments represent the workplaces with predefined application templates and include virtual machines and containers, both emulating network servers. Each environment operates within allocated CPU, memory, disk and traffic resources. An estimated resource allocation depends on a list of usually running applications and their operation hours. However, it is difficult to give accurate a priori workload estimation for a set of virtual environments due to interactive nature of its elements. On the contrary, job requests pool is less interactive and more schedule-specific, which allows to calculate precise preliminary assessments. The first type of workloads is typical to clouds, while the second one is more common to grids. The promising approach to optimal workload balancing in grid-cloud integrations may lie in introduction of market-like behavioristic models.

The economic driven auto-scaling mechanisms for federated clouds bear much resemblance to those acting in real market. These principles should implement policies across network and administrative boundaries, uphold user QoS targets, recognize workload demand patterns, provide compliance with the Service Level Agreement (SLA) contracts, reduce power consumption cost, manage energy-aware placement and live migration of virtual machines between clouds. The key elements of the resource provisioning for federated cloud consist of Cloud Coordinators, Brokers, and an Exchange.

• A Cloud Coordinator exports Cloud services and manages them via market-based trading and negotiation protocols for optimal QoS delivery at minimal cost and energy.
• A Cloud Broker mediates between service consumers and Cloud coordinators.
• A Cloud Exchange acts as a market maker and enables capability sharing across multiple Cloud domains through its match making services.

The detailed description of market-oriented grid resource management is presented in the monograph of $^4$. The approaches to allocate resources in clouds are concerned by $^{22}$. One of recent major IT-problems is the problem of power over-subscription $^{12,19,21,30,39,41}$ manifesting itself as an excessive power supply to massive computational infrastructure leading to significant financial expenses. To address such issue the advanced load balancing techniques adopt various estimators of power used by a virtual machine. The correlations between power consumption and resource utilization are studied in papers $^7,16$. The paper $^{27}$ proposes a novel power measuring technique with performance monitoring counters (PMC) and presents a scheduling algorithm to reduce the measuring errors of recursive power consumption for virtualized resources in large-scale cloud systems. Since the performance and power consumption are mutually exclusive, it is a matter of a trade-off to achieve the optimal load balancing with power capping.
8. Discussion

There is wide acknowledgement that complete resource virtualization is a key to emergence of future Internet architecture. However, the common virtual solutions are susceptible to various attacks compromising or abusing software stacks in cloud environments and targeting security-sensitive data\(^1\). The previous attempts to remedy this problem aimed either at replacement of common virtualization layer\(^17\) with highly customized micro-kernel hypervisors\(^35\) either at protecting hypervisor control flow integrity\(^40\). These approaches to improve security essentially only offered protection from guest VM attacks and did not guarantee safety from hypervisor-level malicious interventions. Moreover, the discarding of virtualization layer is inconsistent with advanced cloud architectures. The study\(^45\) proposes an alternative approach to protect client virtual machines via the concept of nested virtualization decoupling the resource management from security protection in the virtualization layer.

Another isolation problem lies in system-wide oversubscription of resources, especially memory. The respective memory balancing can both result in security breaches\(^37\) and performance degradation\(^42\) for end cloud users. The paper\(^19\) introduces group-based memory deduplication ensuring strict inter-group isolation while achieving intra-group memory efficiency, compared to a system-wide scheme.

Therefore, the obstacles to complete global virtualization originate from both performance and security issues. These remaining problems are mutually dependent, since the tight protection measures consume a lot of resources and performance boost sacrifices efficient security. However, the research is conducted intensively on advanced algorithms and hardware to overcome these remaining obstacles.

9. Conclusions

In this paper, we studied up-to-date considerations of advanced resource virtualization. It was established that both CPU and memory are the key factors providing performance and optimal hardware utilization while being the most complex elements for virtualization. The conducted review states that storage and network components also undergo virtualization, thus affecting the current ISP business models.

In the wake of this theme, we suggested the algorithmic ideas on memory workload estimation in systems with arbitrary nested virtual machines. Our study demonstrates that generalized resource balancing can rely on market-like schemes with agents brokering VM migration to better cloud environment. It is revealed that power balancing also become a matter of interest, since it can reduce financial expenses imposed on distributed systems. We also discussed the topic of recursive virtualization necessary for provisioning performance and security isolation in cloud systems.

10. Acknowledgments

The article is published within applied scientific research performed with financial support of the Ministry of Education and Science of the Russian Federation. Subsidy provision agreement 14.579.21.0010. Universal identifier of the agreement is RFMEFI57914X0010.

11. References

4. Berl A, Fischer A, de Meer H. Using system virtualization to create virtualized networks. Electronic Communications of the EASST. 2009; 17.

5. Bhardwaj S, Jain L, Jain S. Cloud computing: A study of infrastructure as a service (IAAS). International Journal of engineering and information Technology. 2010; 2(1):60–63.

6. Brohi SN, Bamiah MA, Brohi MN, Kamran R. Identifying and analyzing security threats to Virtualized Cloud Computing Infrastructures. In Cloud Computing Technologies, Applications and Management (ICCCTAM). Proceeding of the 2012 International Conference on; 2012 Dec; IEEE; 2012. p. 151–5.

7. Buyya R, Bubendorfer K, editors. Market-oriented grid and utility computing; 75. John Wiley & Sons; 2009.

8. Buyya R, Ranjan R, Calheiros RN. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In Algorithms and architectures for parallel processing. Springer Berlin Heidelberg; 2010. p. 13–31 Available from: http://www.buyya.com/papers/InterCloud2010.pdf

9. Chowdhury NM, Boutaba R. Network virtualization: state of the art and research challenges. Communications Magazine. IEEE; 2009; 47(7):20–6.

10. Dargie W, Schil A. Analysis of the power and hardware resource consumption of servers under different load balancing policies. In Cloud Computing (CLOUD). Proceedings of the 2012 IEEE 5th International Conference on; 2012 Jun. IEEE; 2012. p. 772–8

11. Dillon T, Wu C, Chang E. Cloud computing: issues and
challenges. In Advanced Information Networking and Applications (AINA). Proceedings of the 2010 24th IEEE International Conference on; 2010 Apr. IEEE; 2010. p. 27–33.

12. DoelitzscherF, HeldM, ReichC, Sulistio A. Viteraas: Virtual cluster as a service. In Cloud Computing Technology and Science (CloudCom). Proceedings of the 2011 IEEE Third International Conference on; 2011 Nov; IEEE; 2011. P. 652–7 Available from: http://wolke.hs-furtwangen.de/assets/downloads/CRL-2010-03.pdf

13. DrutskoyD, Keller E, Rexford J. Scalable Network Virtualization in Software-Defined Networks. IEEE Internet Computing. 2013; 17(2):20–7.

14. FeamsterN, GaoL, Rexford J. How to lease the Internet in your spare time. ACM SIGCOMM Computer Communication Review. 2007; 37(1):61–4.

15. FemalME, FreehVW.Safe overprovisioning: Using power limits to increase aggregate throughput. In Power-Aware Computer Systems. Springer Berlin Heidelberg; 2005. p. 150–64

16. FischerA, BoteroJF, Till BeckM, De Meer H, HesselbachX. Virtual network embedding: A survey. Communications Surveys & Tutorials, IEEE. 2013; 15(4):1888–906.

17. JenningsB, Stadler R. Resource management in clouds: Survey and research challenges. Journal of Network and Systems Management. 2014; 1–53.

18. Kalyvianaki E, Charalambous T, HandS. Resource provisioning for multi-tier virtualized server applications. Computer Measurement Group Journal. 2010; (126):6–17.

19. KansalA, Zhao F, Liu J, Kothari N, BhattacharyaAA. (2010, June). Virtual machine power metering and provisioning. Proceedings of the 1st ACM symposium on Cloud computing; 2010 Jun; ACM; 2010. p. 39–50

20. KellerE, SzeferkJ, RexfordjJ, LeeRB. NoHype: virtualized cloud infrastructure without the virtualization. In ACM SIGARCH Computer Architecture News. 2010; 38 (3):350–61.

21. KimS, KimH, LeefJ, JeongJ. Group-based memory oversubscription for virtualized clouds. Journal of Parallel and Distributed Computing. 2014; 74(4):2241–56.

22. KusciD, KephartJO, HansonJE, KandasamyN, Jiang G. Power and performance management of virtualized computing environments via lookahead control. Cluster computing. 2009; 12(1):1–15.

23. Le D, Huang H, Wang H. Understanding performance implications of nested file systems in a virtualized environment. In FAST; 2012. p. 8.

24. Lefurgy C, Wang X, Ware M. Server-level power control. InAutonomic Computing, 2007. Proceedings of the ICAC’07. Fourth International Conference on; 2007 Jun. IEEE; 2007. p. 4–4.

25. LewisPR, FaniyiF, BahsoonR, Yao X. Markets and clouds: Adaptive and resilient computational resource allocation inspired by economics. Taylor & Francis; 2013. Available from: http://www.epics-project.eu/publications/2014_lewis_adaptive.pdf

26. LiuC, MaoY. Inception: Towards a Nested Cloud Architecture. In 5th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 13). USENIX Association.

27. McLaughlinK. Startup Ravello Pitches Nested Virtualization As Way To Solve Tough Cloud Challenges. CRN; 2015 Mar 2. Available from: http://www.crn.com/news/virtualization/300075964/startup-ravello-pitches-nested-virtualization-as-way-to-solve-tough-cloud-challenges.htm

28. MelekhovaA, Markeeva L. Estimating Working Set Size by Guest OS Performance Counters Means. CLOUD COMPUTING. 2015; 48.

29. OceanMJ, BestavrosA, KfouryAJ. snBench: Programming and virtualization framework for distributed multitasking sensor networks. Proceedings of the 2nd international conference on Virtual execution environments; 2006 Jun; ACM; 2006. p. 89–99.

30. PengX, Sai Z. A low-cost power measuring technique for virtual machine in cloud environments. International Journal of Grid & Distributed Computing. 2013; 6(3).

31. PoonWC, MokAK. Bounding the running time of interrupt and exception forwarding in recursive virtualization for the x86 architecture. Technical Report VMware-TR-2010-003, VMware, Inc., 3401 Hillview Avenue, Palo Alto, CA 94303, USA; 2010.

32. ProutA, Kepner J, Michaleas P, Arcand W, Bestor D, Bergeron B, Reuther A. Enabling On-Demand Database Computing with MIT SuperCloud Database Management System. arXiv preprint arXiv:1506.08506; 2015.

33. RaghavendraR, RanganathanP, TalwarV, WangZ, ZhuX. No power struggles: Coordinated multi-level power management for the data center. In ACM SIGARCH Computer Architecture News. 2008; 36 (1):48–59.

34. Rego PAL, CoutinhoEF, GomesD, De SouzaJN. Faircpu: Architecture for allocation of virtual machines using processing features. In Utility and Cloud Computing (UCC). Proceedings of the 2011 Fourth IEEE International Conference on; 2011 Dec; IEEE; 2011. p. 371–376.

35. RimalBP, ChoiE, Lumb I. (2009, August). A taxonomy and survey of cloud computing systems. In INC, IMS and ICDC, 2009. Proceedings of the NCM’09. Fifth International Joint Conference on; 2009 Aug; IEEE; 2009. p. 44–51. Available from: http://eee.guc.edu.ee/Courses/Networks/NETW902 Internet Technology/Research Report/Cloud Computing. pdf

36. RossIFD, de Oliveira IC, De RoseCA, CalheirosRN, BuyyaR. Non-Invasive Estimation of Cloud Applications Performance via Hypervisor’s Operating Systems Counters. ICN 2015; 189.

37. SchaffrathG, WerleC, PapadimitriouP, FeldmannA, BlessR, GreenhalghA, Matuy L. Network virtualization architecture: proposal and initial prototype. Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures; 2009 Aug; ACM; 2009. p. 63–72.

38. Steinberg U, Kauer B. NOVA: a microhypervisor-based secure virtualization architecture. Proceedings of the 5th Eu-
39. Stonebraker M, Madden S, Abadi DJ, Harizopoulos S, Hachem N, Helland P. The end of an architectural era: (it's time for a complete rewrite). Proceedings of the 33rd international conference on very large databases; 2007 Sep; VLDB Endowment. p. 1150–60. Available from: http://bigbe.su/lectures/2014/15.4.pdf
40. Suzuki K, Iijima K, Yagi T, Artho C. Memory deduplication as a threat to the guest OS. Proceedings of the Fourth European Workshop on System Security; 2011 Apr; ACM; p. 1.
41. Tutschku K, Zinner T, Nakao A, Tran-Gia P. Network virtualization: Implementation steps towards the future internet. Electronic Communications of the EASST, 17. 2009.
42. Wang X, Chen M. Cluster-level feedback power control for performance optimization. In High Performance Computer Architecture, 2008. Proceedings of the HPCA 2008. IEEE 14th International Symposium on; 2008 Feb; IEEE; 2008. p. 101–10.
43. Wang Z, Jiang X. Hypersafe: A lightweight approach to provide lifetime hypervisor control-flow integrity. In Security and Privacy (SP). Proceedings of the 2010 IEEE Symposium on; 2010 May; IEEE; 2010. p. 380–95
44. Wang X, Wang Y. Co-con: Coordinated control of power and application performance for virtualized server clusters. In Quality of Service, 2009. Proceedings of the IWQoS. 17th International Workshop; 2009 Jul; IEEE; 2009. p. 1–9.
45. Williams D, Jamjoom H, Liu Y, Weatherspoon H. Overdriver: Handling memory overload in an oversubscribed cloud. In ACM SIGPLAN Notices; ACM. 2011; 46 (7):205–16.
46. Williams D, Jamjoom H, Weatherspoon H. Plug into the Supercloud. Internet Computing, IEEE; 2013; 17(2): 28–34.
47. Xiong P, Wang Z, Jung G, Pu C. (2010, April). Study on performance management and application behavior in virtualized environment. Proceedings of the In Network Operations and Management Symposium (NOMS); 2010 Apr; IEEE; 2010. p. 841–4.
48. Zhang F, Chen J, Chen H, Zang B. CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization. Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles; 2011 Oct; ACM; 2011. p. 203–16.
49. Zhang Q, Cheng L, Boutaba R. Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications. 2010; 1(1):7–18.