Run-time Support for Real-Time Multimedia in the Cloud

Tommaso Cucinotta[1], Karsten Oberle[2], Manuel Stein[2] Peter Domschitz[2], Sape Mullender[3]
[1] Bell Laboratories, Alcatel-Lucent, Dublin, Ireland
[2] Bell Laboratories, Alcatel-Lucent, Stuttgart, Germany
[3] Bell Laboratories, Alcatel-Lucent, Antwerp, Belgium
E-mail: firstname.lastname@alcatel-lucent.com

Abstract

This paper summarizes key research findings in the area of real-time performance and predictability of multimedia applications in cloud infrastructures, namely: outcomes of the IRMOS European Project, addressing predictability of standard virtualized infrastructures; Osprey, an Operating System with a novel design suitable for a multitude of heterogeneous workloads including real-time software; MediaCloud, a novel run-time architecture for offering on-demand multimedia processing facilities with unprecedented dynamism and flexibility in resource management.

The paper highlights key research challenges addressed by these projects and shortly presents additional questions lying ahead in this area.

1 Introduction

The continuous evolution of computation and communication technologies is causing a paradigm shift in our own idea of computing. Indeed, the widespread availability of broadband connections is simply leading to the end of the Personal Computer era, marking the beginning of a new era where computing is mostly distributed. Users not only recur to "the network" to retrieve contents. They also store and manage their data remotely, keeping it accessible from a variety of heterogeneous devices and widespread locations. Users exhibit increasingly challenging requirements on the computing capabilities remotely accessible, not limiting themselves to delegate off-line computations to remote servers, but rather expecting more and more *interactive and real-time* applications to be readily available on-demand. This is witnessed by the increasing use of on-line collaborative document editing or video authoring services, for example.

Being a major driver to the Cloud Computing model, a key role in the new panorama is being played by *virtualization*. With the possibility to host multiple virtualized machines seamlessly onto the same physical hardware, the possibility to create virtual network overlays abstracting away from the actual network topology, and the possibility to dynamically live-migrate virtualized machines while they are running, virtualization technologies constitute an enabler for flexible and efficient management of physical resources in data centers.

However, an application domain where the provisioning of interactive on-line services with nearly "real-time" responsiveness remains challenging from a technical viewpoint is the domain of *multimedia*. Indeed, multimedia contents are characterized by an *isochronous* delivery model, where for example audio or video frames need to be delivered at perfectly regular intervals. However, the network over which most of these contents are distributed nowadays, the Internet, has not been designed with predictability in mind. Furthermore, often multimedia servers that need to deliver contents to many users concurrently make use of software technologies (e.g., Operating System, middleware, etc.) that have been designed for best-effort performance, not for predictable execution. Even more, the use of multimedia compression algorithms leads to a naturally fluctuating networking and computing workload that is usually reflected in variable execution and transmission times. Last, but not least, the use of virtualization technologies increases further the unpredictable behaviors in the execution of services, as due to the increased degree of sharing of physical resources (particularly computing and networking) among different (often heterogeneous) applications. The overall outcome is an *irregular*, *randomly varying* and *unpredictable* delivery of multimedia contents to end users, making it very difficult to adhere to precise QoS specifications in Service Level Agreements (SLAs) [13].

2 Related Work

The problem of guaranteeing stable Quality of Service levels to cloud and distributed applications has been investigated on multiple levels.

The performance implications of data movements have received a lot of attention in the cloud environment, e.g., for proximity reasons [27] and bulk data migration purposes [16]. Placement of computations in large distributed clouds was hypothetically evaluated in [9]. When dealing with deployments spanning geographically distributed data centers, it has been proposed [24] to consider network requirements for the selection of computing locations across the WAN under various scenarios. In [30], authors show the benefits of considering the network topology and overall demand for response times when load-balancing workloads across neighboring data centers. In [6], it is proposed to leverage end-to-end application-level latency expression specifications for optimal placement across geographically distributed locations. In [3], a placement algorithm is proposed that finds a mapping for components of an application with a minimal diameter of the spanned network graph.

Concerning the isolation of virtualized software on the computing level, authors proposed [20] to use an EDF-based scheduling algorithm [21] for Linux on the host to schedule Virtual Machines (VMs). Unfortunately, the proposed scheduler is built into a user-space process (VSched), leading to unacceptable context switch overheads. Furthermore, VSched cannot properly guarantee temporal isolation in presence of a VM that blocks and unblocks, e.g., as due to I/O. IRMOS has improved over these approaches (see Section 3).

Some authors investigated [14] the performance isolation of virtual machines, focusing on the exploitation of various scheduling policies available in the Xen hypervisor [8]. Furthermore, various enhancements to the Xen credit scheduler have been proposed [12] to address various issues related to the temporal isolation and fairness among the CPU share dedicated to each VM. Adaptive CPU allocation has been proposed [23] to maintain a stable performance of VMs, using application-specific metrics to run the necessary QoS control loops.

Concluding, while various solutions have been proposed to the problem of performance isolation in virtualized environments, these are either not focused on critical parameters that are necessary for running real-time applications, or they lack of a proper low-level real-time scheduling infrastructure, which is needed for supporting temporal isolation among concurrently running software components. The following section explains how IRMOS addressed these issues.

3 IRMOS/ISONI Platform

The IRMOS European Project[1] has investigated on how to enhance execution of real-time multimedia applications in distributed virtualized infrastructures. The IRMOS Intelligent Service-Oriented Networking Infrastructure (ISONI) [28, 24] acts as a Cloud Computing IaaS provider, managing and virtualizing a set of physical computing, networking and storage resources available within a provider domain. One of the key innovations introduced by ISONI is its capability to ensure guaranteed levels of resource allocation for individual hosted applications. In ISONI, each distributed application is specified by a Virtual Service Network (VSN), a graph whose vertexes represent Application Service Components (ASCs), deployed as VMs, and whose edges represent communications among them. VSN elements are associated with precise computing and networking requirements. These are fulfilled thanks to the allocation and admission control logic pursued by ISONI for VM instantiation, and to the low-level mechanisms shortly described in what follows. A comprehensive ISONI overview is out of the scope of this paper and can be found in [28, 24].

Isolation of Computing. In order to provide scheduling guarantees to individual VMs scheduled on the same system, processor and core, IRMOS incorporates a deadline-based scheduler [7] for Linux. It provides temporal isolation among multiple possibly complex software components, such as entire VMs. It uses a variation of the CBS algorithm [1], based on EDF, for ensuring that each group of processes/threads is scheduled on the available CPUs

[1] Interactive Real-time Multimedia Applications on Service-oriented Infrastructures. More information is available at: `http://www.irmosproject.eu`.

for a specified time every VM-specific period.

Isolation of Networking. Isolation of the traffic of independent VMs within ISONI is achieved by a VSN-individual virtual address space and by policing the network traffic of each deployed VSN. The virtual addresses overlay avoids unwanted crosstalk between services sharing physical network links. Mapping individual virtual links onto diverging network paths allows for a higher utilization of the network infrastructure by mixing only compatible traffic classes under similar predictability constraints and by allowing selection of more than just the shortest path. Traffic policing avoids that the network traffic going through the same network elements causes any overload leading to an uncontrolled growth of loss rate, delay and jitter for the network connections of other VSNs. It is important to highlight that ISONI allows for the specification of the networking requirements in terms of common and technology-neutral traffic characterization parameters, such as the needed guaranteed average and peak bandwidth, latency and jitter. An ISONI transport network adaptation layer abstracts from technology-specific QoS mechanisms of the networks, like Differentiated Services [5], Integrated Services [32, 31] and MPLS [25]. The specified VSN networking requirements are met by choosing the most appropriate transport network, among the available ones. More detailed information on QoS provisioning between data centers within an ISONI domain is given in [29]. Other interesting results from the research carried out in IRMOS include algorithms for the optimum placement of distributed virtualized applications with probabilistic end-to-end latency requirements [18], a probabilistic model for dealing with workload variations in elastic cloud services [17] and the use of neural networks for estimating the performance of VM execution under different scheduling configurations [19]. The effectiveness of IRMOS/ISONI has been demonstrated, among others, through an e-Learning demonstrator [10].

4 Ongoing and Future Work

The IRMOS project has addressed various challenges in the area of predictable execution of virtualized multimedia applications. However, a number of problems still remain unaddressed. For example, these workloads would benefit from lighter run-time environments than VM instances containing full-fledged OSes, as used in current cloud infrastructures. These are among the motivations of MediaCloud [11] and Osprey [26], two projects from Bell Labs described below.

MediaCloud. Handling the predicted growth of video and media traffic is one of the key challenges future generation networks need to address. Up to now, cache-assisted delivery schemes [15] enabled the networks to scale with the data traffic imposed by video centric services. However, video delivery is becoming more tailored to the specific user accessing it (e.g., user-specific ads). Moreover, future video centric media services will see more people actively producing content. Also, the area of on-line gaming has a growing interest in providing highly dynamic and interactive multimedia. With more contents dynamically produced, customized and accessed from mobile devices, intermediate processing of media streams will need an unprecedented degree of dynamism and adaptability that go beyond the possibilities of today's virtualized infrastructures.

Indeed, the contemporary cloud computing model is based on virtual machines that are statically allocated ahead of time, before it is known who accesses which contents and from where. Furthermore, only relatively small and infrequent adjustments can be done dynamically, as due to the unavoidable "inertia" behind migration of VMs, whose contained OSes often amount to GB of data for the OS volatile memory and tens of GB for the VM disk image. In consequence, today's applications are typically designed in a way, that data has to be moved through the network to where the application is executed [27] which proves costly for live multimedia contents. We believe that this paradigm will change in the future, meaning that an intelligent infrastructure will also force the movement of applications in the line of data and demand sources. Therefore we are working on ways to optimize the delivery of (real-time) media services on top of a distributed cloud environment.

The MediaCloud Project [4] is investigating novel virtualized computing paradigms specifically tied to multimedia applications, where the location of media processing can be quickly altered at run-time, when sources and destinations of the multimedia applications are known. Moving towards a largely distributed service execution paradigm requires software to be split up into fine-grained service components. Designing a service from a plurality of atomic service components requires an online set-up of how those components interact, that is, which media flows the components exchange at service run-time. The customer of such a service should not need to care about the location of execution in the network. MediaCloud takes care of finding best-fit resources during service run-time, when sources and sinks of relevant media streams are known, resulting in reduced end-to-end service latencies and offloaded networks by keeping traffic local. The execution framework ensures fluent media flow forwarding between service components. This deferred allocation puts the foundation for very efficient management of resources. However, one of the main challenges to address is the instantiation of the required media processing functions that needs to be performed so quickly as to not impact the QoE for the end users. The achievement of such a goal is severely obstructed by the use of machine virtualization. Investigations and experiments have shown that using fully-fledged operating systems inside a virtual machine as execution containers can hardly offer the required performance, scalability and efficiency for running distributed real-time media-centric services [4].

MediaCloud introduces a lightweight execution container design, which is fully optimized for supporting efficient execution of fine-grained service components. These can be added and deleted and media flows can be moved between, added to or removed from components at run-time. Such dynamic mechanisms in combination with the ability to move service components between execution resources in the network during run-time, build the basic foundation for an efficient, top-performing and scalable service execution on distributed processing resources in the network.

MediaCloud introduces a novel flow driven execution environment optimized for the processing of media functions, which departs from traditional software stacks being deployed in today's virtualized cloud infrastructures.

Preliminary measurements [11] performed on the prototype implementation proved that MediaCloud is able to provide the envisaged level of agile resource allocation and utilization. It supports instantiation of media processing functions, as well as re-assignment of media processing components across processing resources, in the time-frame of 2 to 3 milliseconds, in some investigated scenarios.

Even highly optimized VM-based systems can accomplish these tasks in seconds but not in milliseconds. Additional investigations indicate that MediaCloud is also able to achieve much more efficient resource utilization. A collection of cooperative media processing tasks executed on a MediaCloud controlled processing resource consumed only about half of the resources needed when doing the same job by making each task a process on the Linux OS. At the same time, we could show significantly better end-to-end service delay figures for a collection of media processing components executed on MediaCloud despite its lower resource utilization.

Osprey. As discussed above, while bringing a number of advantages in terms of ease of (and seamless) management of software, machine virtualization in itself is also constituting the root cause of many technically unnecessary overheads in today's cloud applications. Indeed, virtualized infrastructures have replicated software layers providing similar functions, such as resources management and allocation (e.g., CPU scheduling, memory and peripheral management). Also, many attempts to reduce such overheads so as to obtain a smarter resource management among the hypervisor and the hosted guest OSes usually result in the increase of the degree of para-virtualization of the guest OSes, reducing the advantages of full machine virtualization (e.g., seamless server consolidation and increased isolation/security).

As a consequence, we claim that more attention should be devoted to *OS virtualization* instead, a technique allowing for a single Operating System to create multiple isolated "domains", where independent software can be deployed. For example, the Linux LXC project[2] and FreeBSD Jails[3] provide such a mechanism. However, even though applying QoS-aware (or real-time) resource management techniques in a General-Purpose OS (GPOS) is principally possible, as shown in IRMOS by patching the Linux kernel with a real-time scheduler [7], nonetheless this leads to a suboptimal solution from a number of viewpoints. Still, we keep having replicated functionality among the hypervisor and guest OSes. Furthermore, there are resource wastes due

[2] More information is available at: `http://lxc.sf.net`.

[3] More information is available at: `http://www.freebsd.org/doc/en\_US.ISO8859-1/books/handbook/jails.html`.

to the unawareness of the host and guest schedulers, i.e., in order to guarantee certain real-time performance levels, more resources need to be allocated than strictly needed, because of the hierarchical composition of schedulers [2]. Furthermore, a GPOS is designed for a relatively low number of processes/cores and tasks to handle. However, a big server in a virtualized data center may easily include tens/hundreds of cores in a single machine. A nowadays GPOS does not have the necessary degree of scalability and flexibility in configuration that allow for an efficient management of resources in these conditions.

Osprey [26] is a new OS under development at Bell Laboratories suitable for a multitude of future computing scenarios, including: embedded systems; cloud-hosted real-time multimedia applications with tight timing requirements and highly fluctuating and horizontally scalable resource requirements; future data-intensive and high-performance applications. Osprey includes mechanisms for scalable, low-overhead and energy-aware resources management and scheduling, supporting predictable execution. The OS can be deployed with a very small memory footprint and a lightweight set of functionality, so as to fit within embedded devices dealing with multimedia (e.g., smart phones, set-top boxes, smart TVs, etc...), and very fast boot-up times, so to reduce energy-consumption due to stand-by modes. Osprey can be deployed within network elements, such as base stations, routers, firewalls. In cloud computing environments, Osprey is suitable both for thin clients and for provider-side run-time environments for future cloud applications. It includes OS-level virtualization, and an OS architecture featuring a very small micro-kernel, just capable of switching between address spaces and fielding system calls, traps and interrupts. It uses asynchronous communication primitives among core OS components and for user-kernel space interactions, reducing unneeded overheads. Also, it includes into the core OS mechanisms for check-pointing, migration and recovery of processes, enabling fault-tolerance.

Finally, Osprey integrates Pepys [22], a novel networking protocol for content distribution, with native and efficient support for named replicated contents and mobile users. It also avoids unneeded copies of data across the network stack, enabling high-performance data-intensive applications.

5 Conclusions

In this paper, key research efforts in the area of real-time performance and predictability for multimedia applications in cloud infrastructres have been summarized, along with some of the research challenges that deserve further attention, and a short overview of ongoing research projects promising to address these challenges.

References

[1] L. Abeni and G. Buttazzo. Integrating Multimedia Applications in Hard Real-Time Systems. In *Proceedings of the IEEE Real-Time Systems Symposium*, Madrid, Spain, 1998.

[2] L. Abeni and T. Cucinotta. Efficient virtualisation of real-time activities. In *Proceedings of the IEEE International Workshop on Real-Time Service-Oriented Architecture and Applications (RTSOAA 2011)*, Irvine, CA, December 2011.

[3] M. Alicherry and T.V. Lakshman. Network aware resource allocation in distributed clouds. In *Proceedings of the 31st Annual IEEE International Conference on Computer Communications*, Orlando, Florida, USA, March 2012.

[4] M. Bauer, S. Braun, and P. Domschitz. Media processing in the future internet. In *Proc. of EuroView 2011: Visions of Future Generation Networks*, Wuerzburg, Germany, July 2011.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *RFC2475, An Architecture for Differentiated Service.* IETF, Dec 1998.

[6] F. Chang, R. Viswanathan, and T. L. Wood. Placement in clouds for application-level latency requirements. In *Proceedings of the 5th IEEE International Conference Cloud Computing*, Honolulu, Hawaii, USA, June 2012.

[7] F. Checconi, T. Cucinotta, D. Faggioli, and G. Lipari. Hierarchical multiprocessor CPU reservations for the linux kernel. In *Proceedings of the 5^{th} International Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT 2009)*, Dublin, Ireland, June 2009.

[8] L. Cherkasova, D. Gupta, and A. Vahdat. Comparison of 3 CPU schedulers in Xen. *SIGMETRICS Perform. Eval. Rev.*, 35:42–51, September 2007.

[9] K. Church, A. Greenberg, and J. Hamilton. On delivering embarrassingly distributed cloud services. In *Proceedings of the Seventh ACM Workshop on Hot Topics in Networks (HotNets-VII)*, Calgary, CA, October 2008.

[10] T. Cucinotta, F. Checconi, G. Kousiouris, K. Konstanteli, S. Gogouvitis, D. Kyriazis, T. Varvarigou, A. Mazzetti, Z. Zlatev, J. Papay, M. Boniface, S. Berger, D. Lamp, T. Voith, and M. Stein. Virtualised e-learning on the irmos real-time cloud. *Service Oriented Computing and Applications*, pages 1–16, 2011. 10.1007/s11761-011-0089-4.

[11] P. Domschitz and M. Bauer. Mediacloud - a framework for real-time media processing in the network. In *Proceedings of EuroView 2012*, Wuerzburg, Germany, July 2012.

[12] G. Dunlap. *Scheduler development update*. Xen Summit Asia, Shanghai, 2009.

[13] G. Gallizo, R. Kuebert, G. Katsaros, K. Oberle, K. Satzke, S. Gogouvitis, and E. Oliveros. A service level agreement management framework for real-time applications in cloud computing environments. In *Proceedings of the 2nd International ICST Conference on Cloud Computing (CloudComp 2010)*, Barcelona, Spain, October 2010.

[14] D. Gupta, L. Cherkasova, R. Gardner, and A. Vahdat. Enforcing performance isolation across virtual machines in Xen. In *Proceedings of the ACM/IFIP/USENIX International Conference on Middleware*, pages 342–362, New York,USA, 2006. Springer-Verlag New York, Inc.

[15] M. Hofmann and L. Beaumont. *The book about content networking*. Morgan Kaufman, Feb 2005. ISBN I 55860 834 6.

[16] D. Klein, M. Menth, R. Pries, Phuoc Tran-Gia, M. Scharf, and M. Sollner. A subscription model for time-scheduled data transfers. In *Integrated Network Management (IM), 2011 IFIP/IEEE Internat. Symp. on*, pages 555–562, May 2011.

[17] K. Konstanteli, T. Cucinotta, K. Psychas, and T. Varvarigou. Admission control for elastic cloud services. In *Proc. of the 5th IEEE International Conference on Cloud Computing*, Honolulu, Hawaii, USA, June 2012.

[18] K. Konstanteli, T. Cucinotta, and T. Varvarigou. Optimum allocation of distributed service workflows with probabilistic real-time guarantees. *Service Oriented Computing and Applications.*, 4:68:229–68:243, December 2010.

[19] G. Kousiouris, T. Cucinotta, and T. Varvarigou. The effects of scheduling, workload type and consolidation scenarios on virtual machine performance and their prediction through optimized artificial neural networks. *Journal of Systems and Software*, In Press, Corrected Proof:–, 2011.

[20] B. Lin and P. Dinda. Vsched: Mixing batch and interactive virtual machines using periodic real-time scheduling. In *Proceedings of the IEEE/ACM Conference on Supercomputing*, November 2005.

[21] C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *J. ACM*, 20:46–61, January 1973.

[22] S. Mullender, P. Wolkotte, F. Ballesteros, E. Soriano, and G. Guardiola. Pepys – the network is a file system. Technical Report TR RoSaC20114, Bell Labs and Rey Juan Carlos University, 2011.

[23] R. Nathuji, A. Kansal, and A. Ghaffarkhah. Q-Clouds: Managing Performance Interference Effects for QoS-Aware Clouds. In *Proceedings of the 5th European Conference on Computer systems (EuroSys)*, Paris, France, April 2010.

[24] K. Oberle, M. Kessler, M. Stein, T. Voith, D. Lamp, and S. Berger. Network virtualization: The missing piece. In *Intelligence in Next Generation Networks, 2009. ICIN 2009. 13th International Conference on*, pages 1–6, October 2009.

[25] E. Rosen, A. Viswanathan, and R. Callon. *RFC3031, Multi-protocol Label Switching Architecture*. IETF, Jan 2001.

[26] J. Sacha, J. Napper, H. Schild, S. Mullender, and J. McKie. Osprey: Operating system for predictable clouds. In *in Proceedings of the 2nd Workshop on Dependability of Clouds, Data Centers and Virtual Machine Technology (DCDV'12)*, Boston, MA, USA, June 2012.

[27] B. Tiwana, M. Balakrishnan, M. Aguilera, H. Ballani, and Z. M. Mao. Location, location, location! modeling data proximity in the cloud. In *In HotNets IX: Ninth Workshop on Hot Topics in Networking*, pages 1–6, Monterey, CA, October 2010.

[28] T. Voith, M. Kessler, K. Oberle, D. Lamp, A. Cuevas, P. Mandic, and A. Reifert. *ISONI Whitepaper v2.0*, 2009.

[29] T. Voith, K. Oberle, and M. Stein. Quality of service provisioning for distributed data center inter-connectivity enabled by network virtualization. *Elsevier Future Generation Computer Systems (FGCS 2011)*, 2011.

[30] I. Widjaja, S. Borst, and I. Saniee. Geographically distributed datacenters with load reallocation. In *DIMACS Workshop on Cloud Computing*, Piscataway, NJ, USA, December 2011.

[31] J. Wroclawski. *RFC 2210, The Use of RSVP with IETF Integrated Services*. IETF, Sep 1997.

[32] J. Wroclawski. *RFC2211, Specification of the Controlled Load Quality of Service*. IETF, Sep 1997.