A CLOUD PLATFORM FOR REAL-TIME INTERACTIVE APPLICATIONS

Andreas Menychtas, Dimosthenis Kyriazis, Spyridon Gogouvitis
National Technical University of Athens, Iroon Polytechniou 9, Zografou Campus, 15773 Athens, Greece

Karsten Oberle, Thomas Voith
Alcatel-Lucent Bell Labs, 70435 Stuttgart, Germany

Georgina Galizo, Soeren Berger
HLRS - High Performance Computing Center Stuttgart, Nobelstrasse 19, 70569 Stuttgart, Germany

Eduardo Oliveros
Telefónica I+D, Emilio Vargas 6, 28043 Madrid, Spain

Michael Boniface
IT Innovation Centre, University of Southampton, 2 Venture Road, SO16 7NP, Southampton, U.K.

Keywords: Cloud computing, Cloud architecture, Cloud platform, Software as a service, Platform as a service, Infrastructure as a service, Real-time, Service oriented.

Abstract: Cloud Computing is considered nowadays as the future of ICT systems leveraging new methodologies for developing, providing and consuming services. Even though many people believe that *"Cloud"* is just another buzzword for utility computing, this new computing paradigm is not only changing the design of modern computing platforms in technical level, but it also impels, from the market perspective, the creation of new value chains and business models. Beyond the great advantages of cloud technologies for scalability, elasticity and low operational cost, there are still many technical complexities and limitations on provisioning and management of applications with high QoS demands that disallow the wide adoption of cloud solutions. In this paper we present a novel cloud platform capable to support real-time interactive applications considering their full lifecycle including service engineering, SLA negotiation, provisioning and monitoring. This platform has been designed and implemented consolidating management and control of the infrastructure and services at all points in the value chain to support real-time interaction focusing on its business and commercial orientation.

1 INTRODUCTION

Although cloud computing (Buyya, 2009) as another distributed computing paradigm is not something new, nowadays seems that the number of people and organizations exploiting the cloud computing capabilities is increasing. The main IT players such as Google and Microsoft have already developed platforms to offer cloud services hosted in their datacenters and at the same time hundreds of new companies worldwide are involved in the service delivery value chain either by using their owned infrastructures or by providing added value services. The new cloud ecosystems are changing the way the computing, storage and networking resources are purchased and consumed creating new business paradigms and value networks for the service

Menychtas A., Kyriazis D., Gogouvitis S., Oberle K., Voith T., Galizo G., Berger S., Oliveros E. and Boniface M..
A CLOUD PLATFORM FOR REAL-TIME INTERACTIVE APPLICATIONS.
DOI: 10.5220/0003387003970403
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 397-403
ISBN: 978-989-8425-52-2

delivery. In contrast with the proprietary software where the license schemas are rather simple, the cloud based services -exploiting the advantages of the cloud for scalability, elasticity, multi-tenancy and reliability- are strongly related with the business aspects of the application and platform influencing all processes of the service lifecycle.

From the technical perspective and based on the SPI (Service - Platform - Infrastructure) model (NIST, 2009), the cloud solutions can be categorized in three main classes:

- **Infrastructure as a Service (IaaS)**
- **Platform as a Service (PaaS)**
- **Software as a Service (SaaS)**

In the paper we present a novel cloud platform, which was developed in the frame of the EU-funded project IRMOS (see References section) targeting soft real-time applications that have stringent timing and performance requirements. This platform combines Service Oriented Infrastructures - SOIs (Erl, 2005) with virtualisation technologies to manage and provision computational, storage and networking resources as well as to communicate with legacy systems such as wifi locators. Service-oriented design has several advantages that the proposed solution exploits in order to dynamically connect people, processes, information and services that are spanning across different domains or layers of the platform architecture. The platform specification advances existing service-oriented approaches by providing methodologies, tools and mechanisms in order to efficiently operate, manage and reconfigure services and resources under *real-time constraints*. The constraints are expressed as Quality of Service (QoS) terms in Service Level Agreements (SLAs) that are dynamically negotiated and define commitments between the different stakeholders in the value chain.

Provisioning applications in virtualised infrastructures with guaranteed QoS is a non-trivial task. At the core is the need to intelligently allocate and adapt resource provisioning policies based on knowledge of the application, customer and infrastructure while during the operational phase the ability to monitor the actual performance triggering mitigating actions in real-time to overcome from exceptional behaviour of services and resources. The platform supports these requirements through a set of Framework Services - FS that implement a QoS-oriented Service Engineering methodology (Kyriazis, 2010) linking the lifecycle processes with novel modelling tools and autonomic management services. In that sense, the platform is not only a set of interacting services but also a real-time enabled system in which instances of services with real-time capabilities are deployed in the virtual environments supervising the application lifecycle and guaranteeing the agreed QoS level.

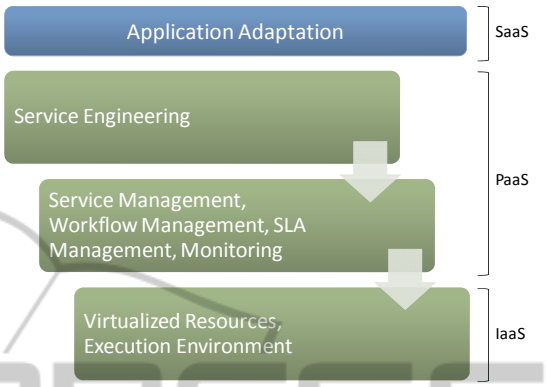

Figure 1: Platform Layers based on SPI Cloud Model.

Following the SPI cloud model (Figure 1) the overall management processes of the platform as well as the individual services are designed, developed and distributed in order to support real-time interactivity not only for the applications but also for the infrastructure itself. The platform provides semantic representations of systems in order to efficiently allow for scalability, interoperability and adaptation and therefore solves several problems in reference to QoS provisioning such as the real-time scheduling of the service execution and mapping of the application workflows and requirements to low level resource parameters. To achieve the guaranteed real-time end-to-end performance, the platform continuously exchanges management information between its two main building blocks the Framework Services - FS (PaaS layer) and the Intelligent Service Oriented Network Infrastructure - ISONI (IaaS layer) that are described in detail in next sections.

The remainder of the paper is structured as follows: Section 2 describes the methodology and the principles that were followed in the design of the platform architecture to achieve real-time QoS provision while section 3 introduces the control loops concept that allows the infrastructure to provide real-time QoS guarantees and the overall platform architecture design as well as the specifications of the FS and ISONI are detailed in section 4. The paper concludes with a discussion on future research and potentials for the current study.

2 REAL-TIME QOS PROVISION IN CLOUDS

The primary objective of the proposed platform is to develop a cloud solution capable to support QoS guarantees through all layers of the system for interactive real-time multimedia applications. The architecture considers the full service lifecycle of both service-based systems and legacy applications deployed on cloud resources including service engineering, service level agreement design and resource management and monitoring. QoS parameters at application, platform and infrastructure levels are given specific attention as the basis for dynamic QoS provisioning in real-time.

The research work around QoS provisioning and SLA enforcement in clouds is not something new. State of the Art cloud providers such as Amazon EC2 have already implemented mechanisms for monitoring the guarantees for QoS which are mainly expressed as "Annual Uptime Percentage" enriched with penalty models (Amazon EC2-SLA). This methodology is generic and static without taking into consideration the particular QoS requirements of each application and cannot be adapted during runtime. Emeakaroha et al. (Emeakoraha, 2010) in the LoM2HiS approach and Stantchev et al. (Stantchev, 2009) propose solutions for mapping the high level application requirements to the monitored metrics focusing mainly on SLA monitoring and SLA violation detection for achieving QoS provisioning. Nevertheless these solutions do not exploit any application or platform reconfiguration techniques so as to maintain the overall system stability and acceptable operational level as required by real-time interactive applications. On the other hand the real-time clouds approaches have been discussed from Liu et al. (Liu, 2010) and Sarathy et al. (Sarathy, 2010) focusing only on resource management at infrastructure level.

The proposed platform architecture supporting real-time service, human and resources interactivity implements the following key features:

- *Real-Time QoS Specification*
- *Event Prediction*
- *Dynamic SLA (Re)Negotiation*
- *On-Demand Resource Provisioning*
- *QoS aware Event Monitoring*

Service-oriented design principles are considered as an important aspect of the architecture throughout all cloud platform layers. The platform adopts a service-oriented approach to allow services interact dynamically and continuously even though they span between different domains, from the application layer to the layer of network resources and execution environment. The challenge is to carefully design and synchronize this rich set of services so as to efficiently operate, manage and reconfigure all resources under real-time conditions, providing to the end users and to the associated applications the appropriate and required Quality of Service. All QoS terms are dynamically negotiated and agreed in SLAs between the various actors of the value chain (Gallizo, 2009) taking into consideration the QoS guarantees from both application and resource perspectives. All platform and infrastructure capabilities are offered as on-demand services, although the architecture design of the media applications may vary from traditional n-tier enterprise applications to service-oriented workflows. In that frame, the service orchestrations and processes are developed in a way so as to preserve the real-time attributes throughout the whole infrastructure layers.

A major challenge for SaaS providers wanting to exploit the benefits of cloud computing is to manage QoS commitments to customers throughout the lifecycle of a service. The PaaS offers to SaaS providers services tools for estimating resource needs in advance of execution and mechanisms for negotiating QoS with service providers and provisioning virtualised resources. This also includes assessing the probable technical and economic outcomes of provisioning policies and management actions even if the application or resources do not perform as expected or need to be adjusted. The proposed approach considers analysis and decision support within temporal and business constraints to determine which actions are triggered offline (i.e. pre-execution) or online (i.e. during execution). Because faults are inevitably going to occur, strong fault detection and recovery mechanisms are implemented. This can have a great impact on the real-time capabilities of the platform, since intelligent fault recovery mechanisms allow timing constraints to still be met in case of a failure. The performance of the monitoring and control loops between cloud layers, described in section 3, is as essential factor in ensuring that QoS guarantees are maintained.

At the IaaS layer, real-time functionality is supported by the Intelligent Networking and the Execution Environment infrastructures applying virtualization techniques for several types of resources such as networking, storage and computational. The IaaS layer instantiates, manages and monitors the various resources according to the

set of services that are deployed. To this direction, Execution Environment considers multitasking, threads with priorities and an appropriate number of interrupt levels to achieve QoS objectives.

Another essential element of cloud computing, especially of PaaS layer, is the ability to deliver on-demand services with minimal manual configuration. In that sense, all platform subsystems can be self-managed and reconfigured in order to achieve management efficiencies, to react to QoS failures (such as for instance an SLA violation or Network link failure) in a timely way and avoid the escalation of interlayer problems.

Cloud utilisation involves several processes that span in different cloud layers and stakeholders. For example, the platform supports application developers in engineering their applications for the cloud implementing standard specifications and methodologies, while other processes support application provisioning and execution through the innovative virtualised execution environment and networking infrastructure. Therefore, the platform does not only provide a set of services but also cross layer workflow mechanisms that consider the control channels and information exchanges which are required to support real-time management of interactive applications throughout the full lifecycle.

3 CONTROL LOOPS

In order to provide QoS guarantees for interactive real-time multimedia applications, platform provides a set of services and cross layer workflows that consider the control channels and information exchanges which are required to support real-time management throughout the full lifecycle. All subsystems are self-managed and reconfigured in order to achieve management efficiencies, and to react on QoS failures (such as an SLA violation or network link failure) in a timely way. To achieve this, we introduce three control loops at infrastructure level providing the necessary functionality in order to maintain QoS metrics across the architectural levels. The **IRMOS Control Loops** are the following and are depicted in Figure 2:

- *Application Control:* It deals with the relationship between users and applications required to guarantee the application QoS. This control loop is managed by the application itself in response to either user events or platform events. It is implemented with the use of models, workflows and tools that produce artifacts capturing the applications' behavior and estimating resource needs in advance of execution. During runtime it refers to application monitoring that may for example trigger events or require for changes in the provided resources.
- *Environment Control:* It deals with the relationship between applications and virtual resources in order to guarantee the platform QoS, as agreed in the SLAs. This control loop is managed by the platform services in response to application and virtualisation events. It is implemented by the framework services that support and manage the applications at run-time (e.g. actions triggered if either the application or resources do not perform as expected or need to be adjusted).
- *Virtualization Control:* It deals with the relationship between virtual and physical resources in order to guarantee the infrastructure QoS. This control loop is managed within IaaS layer called ISONI in response to platform or physical events. It is implemented by intelligent networking mechanisms as well as by the real-time enabled execution environment for computational and data storage services.

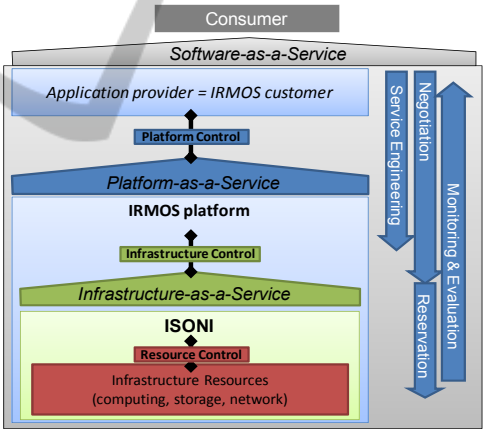

Figure 2: Platform Control Loops.

We identified five (5) main processes / channels implementing the control loops:

- Service Engineering
- Negotiation / Renegotiation
- Reservation
- Monitoring and Evaluation

The actual implementation of the control loops refers to tools and services used on different levels in order to monitor the applications' execution, communicate possible events and take corrective actions if needed. These mechanisms are analyzed in the following section as part of the platform architecture.

4 PLATFORM ARCHITECTURE

In this section we describe the overall architecture of the platform and its main subsystems. The high-level view of the platform architecture is shown in the following figure:

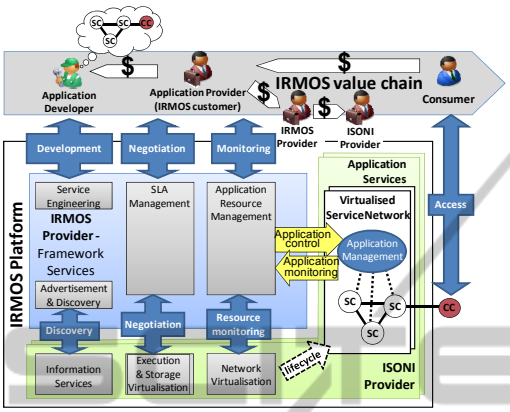

Figure 3: Platform Overall Architecture.

As indicated in Figure 3, the platform has the two main building blocks: PaaS (Framework Services) and IaaS (ISONI). During the architecture design and specification we followed an innovative approach on how these blocks will interact, and in that sense their relation is considerably different of the conventional SOA or Cloud platforms because of its real-time orientation, the virtualization capabilities and the way the management information is shared between platform and infrastructure layers to ensure end-to-end QoS. Initially, the Framework Services provide service engineering tools for the application developer and provisioning services for the IRMOS provider as the entity responsible for offering applications.

Each VSN is instantiated by ISONI and includes particular technical requirements defined by the application developer at design time and specific QoS customisation defined by the customer at runtime. These requirements are relayed to ISONI during the SLA negotiation through the Framework Services. ISONI cannot be accessed directly by end users (Customers or Consumers) and their access privileges are limited to application service components.

As already mentioned, the Framework Services communicate continuously and in various ways with ISONI. Each ISONI provider advertises its capabilities to Framework Services so as to be discovered later and as second step negotiate SLAs for an application. Additionally ISONI provides monitoring data and notification events for each VSN (at the SC level) to Framework Services that are used for both runtime (control) and design time (development and modelling). The fact that real-time functionality is required on some components of the Framework Services layer demands that instances of these components will be deployed and run in the VSN where the real-time QoS is guaranteed. As presented in Figure 3, the core platform services in PaaS interact with service instances running in VSNs for controlling and monitoring the application during application execution.

The Execution Environment and the Intelligent Networking subsystems are architecturally close and are expected to communicate continuously during all the processes of the platform (Oberle, 2010). These subsystems are wrapped in the ISONI infrastructure. The main objective of this layer is to virtualize resources, provision of application services and monitor the resources without the need for knowledge on the application itself. The Execution Environment subsystem, considered as an enhanced virtualization platform, includes the storage systems and is implemented so as to address the QoS and especially the real-time requirements of the application services. The network resources, provided through a VPN like approach, are classified and advertised to the Framework Services in QoS classes.

4.1 Framework Services

The Framework Services – FS is the layer between applications and virtualized resources offered by IaaS providers. This layer corresponds to the PaaS layer of SPI cloud model and the architecture is shown in Figure 4. The architecture consists of two main elements, Service Engineering and Service Management.

The Framework Services layer aims to provision and manage the execution of real-time services on request of the Application Layer using virtualized resources. These resources are offered by the IaaS providers conforming to the real-time constraints as determined in the application SLAs. Apart from the execution of the services that are provided to customers, Framework Services support service engineering, fully automated SLA (re-)negotiation, mapping of high level performance parameters to low level resource parameters, discovery and reservation of the virtualized resources needed for the execution of an application. In the execution phase of the application, FS monitor continuously and manage the application components and the resources either directly, through the application

wrappers based on predefined application specific policies, or relaying the management requests to ISONI layer based on operational policies of the platform. It should be noted that instances of the Framework Services such as Workflow Enactor and Monitoring Services are deployed within the application VSN so as to benefit from the QoS provisions the IaaS can offer.

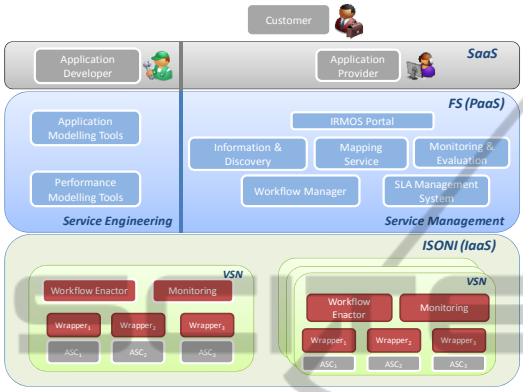

Figure 4: Framework Services General Architecture.

For the communication with the user of the platform, the *IRMOS Portal* component has been implemented, which provides the necessary interface to enable the end-user of the application to request the SLAs templates, invoke the negotiation process and the reservation of virtualized resources on the IaaS layer. In addition, FS functionality includes starting and stopping of an application execution relaying the requests, through the Service Management system, to the appropriate application service components running in a VSN.

4.2 Intelligent Service Oriented Network Infrastructure

ISONI (Intelligent Service Oriented Network Infrastructure) is an IaaS environment, consisting of a network of resources (e.g. CPU, storage, software, etc) managed and controlled by a middleware, which allows resource sharing among multiple services (ISONI White Paper, 2010), (Oberle, 2009), (Oberle, 2010). The general idea is to provide QoS capable infrastructure resources on demand for dynamically deployed services. As already described in a previous section, a service is usually composed out of several smaller and simpler services, in the following called Service Components (SC). ISONI is agnostic to services, thus the decomposition of services into SC is not its responsibility, and is accomplished by the Framework Services layer. The objective of ISONI is to provide these SCs with the best resources (Execution Environments and network links). Figure 5 depicts the main components of the IaaS layer.

ISONI exposes its virtualized infrastructure in the form of VSNs, which can be seen as a graph whose vertexes are the SCs and whose edges are the Virtual Links. In the proposed platform the Framework Services will state their infrastructure requirements using a VSN description.

It is the role of the Framework Services layer, which is application aware, to decompose its Services and Applications into Services Components, build the VSN description and request resources from ISONI which is application unaware. The VSN description is transferred to the IaaS layer with the request to instantiate the service. Then, the ISONI has to automatically and autonomously map the highly abstracted resource request in form of the VSN description onto the network of real resources, to deploy the components in tailored execution environments on suitable resources, and to interlink them while observing QoS requirements. This instantiated VSN builds an independent layer 3 overlay network, i.e., there is no limitation on the L3 protocol stack used by the SCs.

The ISONI architecture is composed of functional blocks, where each takes on a different task for the management of the ISONI resources and deployed VSNs.

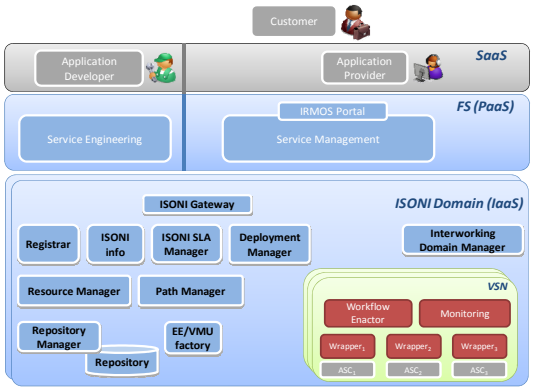

Figure 5: ISONI General Architecture.

Figure 5 shows the functional building blocks of the ISONI management. The interfaces to the Framework Services and tools are indicated on top. The functional blocks Resource Manager, Storage Manager and Path Manager would be deployed in a two-level management architecture based on the composite structure, *Domain level* and *Node level*. The resource responsibility lies with the Node level, i.e. the Node control and resource reservations are maintained by the middleware functional blocks

running at Node level, whereas the Domain level instances coordinate the ISONI Nodes. This approach guarantees efficient management of the VSNs as well as the resource scalability, a key requirement in cloud environments.

5 CONCLUSIONS

The paper presented a novel cloud platform capable to support the full lifecycle of applications with real-time QoS requirements. In addition to the platform design and specification we described methodologies and best practices that have been followed in the platform to effectively provision and manage application services and infrastructure resources during runtime. The proposed platform promises to significantly advance the state-of-the-art in provisioning applications with guaranteed QoS on virtualised infrastructures and has been validated by three different application scenarios, which were also the basis for the requirements identification during the design process. The evaluation results were impressive in all scenarios with the system capable to reconfigure in an acceptable time frame. In cases of live migration of virtual machines the reconfiguration time (from the user perspective) is close to one second, depending on the application, while in cases of user or system driven SLA renegotiation the platform scaling is completed in less than a minute. The final prototype supporting service resilience and events evaluation processes advancing further the QoS provisioning and real-time management capabilities of the platform was released on January 2011 and is available on project website.

ACKNOWLEDGEMENTS

The research leading to these results has been performed in the context of the project *"Interactive Real-time Multimedia Applications on Service Oriented Infrastructures"* (IRMOS). The project has received funding from the EC Seventh Framework Programme FP7/2007-2011 under grant agreement n° 214777.

REFERENCES

Amazon EC2 Service Level Agreement: http://aws.amazon.com/ec2-sla.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. 25, 6 (Jun. 2009), 599-616.

Emeakaroha, Vincent C.; Brandic, Ivona; Maurer, Michael; Dustdar, Schahram; , "Low level Metrics to High level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments," High Performance Computing and Simulation (HPCS), 2010 International Conference on , vol., no., pp.48-54, June 28 2010-July 2 2010.

EU IST IRMOS Project, http://www.irmosproject.eu

G. Gallizo et al., "A Service Level Agreement Management Framework for Real-time Applications in Cloud Computing Environments", CloudComp 2010, Barcelona, Spain, 25.10-28.10.2010.

Georgina Gallizo et al., "Service Level Agreements in Virtualised Service Platforms", eChallenges, Warsaw, 21-23 October 2009.

IRMOS Project ISONI Whitepaper V2.0, ALUD and USTUTT, July 2010. Available on-line at http://www.irmosproject.eu

Kyriazis D, Einhorn R, Furst L, Braitmaier M, Lamp D, Konstanteli K, Kousiouris G, Menychtas A, Oliveros E, Loughran N, Nasser B, "A Methodology for engineering real-time interactive multimedia applications on Service Oriented Infrastructures", IADIS Applied Computing 2010, Timisora, Romania, 2010.

NIST Definition of Cloud Computing, Peter Mell and Tim Grance, Version 15, 2009, available online here: http://csrc.nist.gov/groups/SNS/cloud-computing

Oberle, K., Voith, T., Stein, M., Gallizo, G., Kübert, R., "The Network Aspect of Infrastructure-as-a-Service", ICIN2010, Berlin, October 2010.

Oberle, L., Kessler, M., Voith, T., Stein, M., Lamp, D., Berger, S., "Network Virtualization: The missing piece", ICIN2009, Bordeaux 26-29.10.09.

Sarathy, V.; Narayan, P.; Mikkilineni, R.; "Next Generation Cloud Computing Architecture: Enabling Real-Time Dynamism for Shared Distributed Physical Infrastructure," Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on , vol., no., pp.48-53, 28-30 June 2010.

Shuo Liu, Gang Quan, Shangping Ren, "On-Line Scheduling of Real-Time Services for Cloud Computing," services, pp.459-464, 2010 6th World Congress on Services, 2010.

T. Erl, "Service-oriented Architecture: Concepts, Technology, and Design", Upper Saddle River: Prentice Hall PTR, ISBN 0-13-185858-0, 2005.

Vladimir Stantchev and Christian Schriipfer, Negotiating and Enforcing QoS and SLAs in Grid and Cloud Computing," Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing, pp. 25-35, April 2009.