

Chapter 4.12

Programming Interfaces for Realtime and Cloud- Based Computing

Gregory Katsaros

National Technical University of Athens, Greece

Tommaso Cucinotta

Scuola Superiore Sant'Anna, Italy

ABSTRACT

Research in the fields of Grid Computing, Service Oriented Architectures (SOA) as well as virtualization technologies has driven the emergence of Cloud service models such as Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS).

The appearance of different business roles according to this classification, potentially with differing interests, introduces new challenges with regard to the tools and mechanisms put in place in order to enable the efficient provisioning of services. Security, Quality of Service (QoS) assurance, and real-time capabilities are just a few issues that the providers are trying to tackle and integrate within the new products and services that they offer. In this chapter, we make an overview of the approaches that aim to APIs for real-time computing. In the first part of this chapter, several Real-Time Application Interfaces will be presented and compared. After that, we will document the state-of-the-art regarding the Cloud APIs available and analyze the architecture and the technologies that they support.

INTRODUCTION

New generation interactive distributed applications, such as various multimedia, virtual collaboration and e-learning applications, have

significant demands on processing, storage and networking capabilities, as well as stringent timing requirements. For these applications, the time at which individual computations and data transfers are terminated is as important as their functional correctness. Contrarily to the traditional domain of hard real-time systems, where a single violation of

DOI: 10.4018/978-1-61350-456-7.ch4.12

the timing constraints is not acceptable because it would lead to potential overall system failures and/or life losses, the mentioned applications possess instead soft timing constraints, whose violation leads to degradation of the offered Quality of Service and interactivity level.

Soft real-time applications, and especially multimedia ones, would greatly benefit from a real-time run-time support like commonly available on Real-Time Operating Systems. In fact, this type of OS usually provides those features that allow for a well-known, predictable and analyzable timing behavior of hosted applications: all kernel segments are characterized with well-known worst-case durations, scheduling latencies and interrupt latencies which may be controlled through the appropriate tuning of the interrupt and process schedulers. A set of real-time scheduling policies is available for the system designer and programmers. Time may be measured (and timers may fire) with a high precision (typically sub-millisecond) while there are tools available for WCET estimation and (off-line) analysis, as well as for schedulability analysis. Unfortunately, such OSes are designed for embedded control applications, thus they imply serious constraints on the supported hardware and available high-level software infrastructures.

Nowadays multimedia applications are increasingly complex and they tend to be distributed, thus their development on a hard real-time OS may be overly prohibitive, due to the lack of such OS functionality as: compression libraries, the support for a wide range of multimedia devices and protocols, including the possibility to handle various types of media, and the availability of a complete networking stack.

On the other hand, General-Purpose Operating Systems (GPOS) constitute the ideal development platform for multimedia applications. Unfortunately, GPOSes are not designed to provide the run-time support that is necessary for meeting timing requirements of individual applications. Therefore, in recent years, various efforts have

been done towards the integration of real-time technologies within GPOSes, particularly within Linux, for the availability of its kernel as open-source and its worldwide diffusion. Various drawbacks make the Linux kernel, as well as most of the GPOSes, particularly unsuitable for running real-time applications: the monolithic structure of the kernel and the wide variety of drivers that may be loaded within, the impossibility to keep under control all the non-preemptable sections possibly added by such drivers, the general structure of the interrupt management core framework that privileges portability with respect to latencies, and others. As a result, the latency experienced by time-sensitive activities can be as large as tens or hundreds of milliseconds. This makes Linux non-suitable for hard real-time applications with tight timing constraints. Though, soft real-time applications may run quite well within such an environment, especially when the original kernel is modified so to integrate the necessary real-time capabilities.

Furthermore, research in the fields of Grid Computing, Service Oriented Architectures (SOA) as well as Virtualization technologies has driven the emergence of Cloud service models such as Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). Moreover, the emergence of different business roles according to this classification, potentially with differing interests, introduces new challenges with regard to the tools and mechanisms put in place in order to enable the efficient provisioning of services. Security, Quality of Service (QoS) and efficiency are just a few issues that the providers are trying to tackle and integrate within the new products and services that they offer. In general, the SaaS provider offers an application as a service over a distributed environment to the end users. The PaaS provider offers a development platform and environment consisting of services and storage units hosted on the cloud, while the IaaS provider offers resources on demand depending on each deployed instance. The

15 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/programming-interfaces-realtime-cloud-based/62491

Related Content

The Mediation Role of Knowledge Sharing Between Organizational Learning and Technological Innovation Practice

Zhimin Wang and Choon Ling Kwek (2020). *Disruptive Technology: Concepts, Methodologies, Tools, and Applications* (pp. 601-624).

www.irma-international.org/chapter/the-mediation-role-of-knowledge-sharing-between-organizational-learning-and-technological-innovation-practice/231208

Monitoring and Auditing in the Cloud

Indira K. and Vennila A. (2018). *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 856-882).

www.irma-international.org/chapter/monitoring-and-auditing-in-the-cloud/203538

Design Features of High-Performance Multiprocessor Computing Systems

Gennady Shvachykh, Nina Rizun, Olena Kholod, Olena Ivaschenko and Volodymyr Busygin (2019). *Cases on Modern Computer Systems in Aviation* (pp. 381-401).

www.irma-international.org/chapter/design-features-of-high-performance-multiprocessor-computing-systems/222197

A Structured Method for Security Requirements Elicitation Concerning the Cloud Computing Domain

Kristian Beckers, Isabelle Côté, Ludger Goeke, Selim Güler and Maritta Heisel (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 782-805).

www.irma-international.org/chapter/a-structured-method-for-security-requirements-elicitation-concerning-the-cloud-computing-domain/192901

Providing Automated Holistic Process and Knowledge Assistance During Software Modernization

Gregor Grambow, Roy Oberhauser and Manfred Reichert (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 351-395).

www.irma-international.org/chapter/providing-automated-holistic-process-and-knowledge-assistance-during-software-modernization/192885