



A distributed modular platform for the development of cloud based applications



G. Fylaktopoulos^a, M. Skolarikis^a, I. Papadopoulos^a, G. Goumas^b, A. Sotiropoulos^c, I. Maglogiannis^{d,*}

^a B-Open S.A., Laskaraton 11A Thessaloniki Pylaia 54250, Greece

^b Computing Systems Laboratory, National Technical University of Athens, Greece

^c GRNET S.A., Kifisias Av. 7, 11523, Athens, Greece

^d Department of Digital Systems, University of Piraeus, Greece

HIGHLIGHTS

- A cloud IDE that supports Model Driven Development.
- Component-based application development and multilayer programming.
- Automatic database connectivity.
- Internal model and functionality repository are supported.
- Evaluation and comparison with existing systems.

ARTICLE INFO

Article history:

Received 1 April 2016

Received in revised form

26 December 2016

Accepted 20 February 2017

Available online 22 February 2017

Keywords:

Cloud based development
Model Driven Development
Cloud Integrated Development
Environment

ABSTRACT

In this paper we describe the CIRANO platform, a modular Integrated Development Environment (IDE) for cloud based applications. The proposed platform is built to support Model Driven Development (MDD) and team collaboration, facilitating the rapid development of advanced applications in the cloud. The paper presents at a first stage the state of the art in the field of cloud IDEs and describes the design, implementation and technical details of the CIRANO platform. The main features of the proposed platform are presented in two case studies concerning the development of an application from scratch and porting of an existing application. The paper discusses the findings in comparison with existing tools and proposes extensions of the platform as future work.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Nowadays, cloud computing is recognized as a dominant computing model in IT infrastructures, enabling flexible, ubiquitous, on-demand and cost-effective access to a wide pool of shared resources [1]. It is generally noticed that large and diverse organizations have adopted the cloud paradigm for their information systems, enjoying the following two key features: low costs by releasing them from the burden to invest on hardware infrastructures and software licenses, and reduced operational complexity,

as organizations are able to focus on the quality of their products and services rather than on the management of complex IT systems [2]. Furthermore, scale economy drives additional decrease in total cloud infrastructure costs, as numerous small-scale and typically underutilized data centers are replaced by larger infrastructures that target higher resource efficiency. On the application side, large families of applications including desktop, business and entertainment, have found their way to the cloud creating a demanding and fast evolving new ecosystem [3].

The proliferation of the cloud paradigm has created a strong trend to transfer traditional services and applications to the cloud. In this context, software development environments are a critical application domain that has also gained significant popularity through its “cloudified” versions. Transferring major services and applications to the cloud has created new demands for productive software development. Cloud concepts and technologies provide a valuable substrate to support software development environments

* Correspondence to: University of Piraeus, Grigoriou Lampraki 126 Piraeus, Greece. Fax: +30 2104142517.

E-mail addresses: gfylak@b-open.gr (G. Fylaktopoulos), mikeskolari@b-open.gr (M. Skolarikis), ipap@b-open.gr (I. Papadopoulos), goumas@cslab.ece.ntua.gr (G. Goumas), sotirop@grnet.gr (A. Sotiropoulos), imaglo@unipi.gr (I. Maglogiannis).

List of Acronyms

aPaaS	Application-platforms-as-a-service
CCIM	CIRANO CIM
CIM	Computational Independent Model
CPIM	Cirano PIM
CPSM	Cirano PSM
CSS	Cascading Style Sheets
DNS	Domain Name System
HAProxy	High Availability Proxy
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
LAN	Local Area Network
MDA	Model Driven Architecture
MDD	Model Driven Development
MVC	Model View Controller
OOP	Object-Oriented Programming
PIM	Platform Independent Model
PSM	Platform Specific Model
REST	Representational State Transfer
RPC	Remote Procedure Call
SOAP	Simple Object Access Protocol
UML	Unified Modeling Language
VPN	Virtual Private Network
WebRTC	Web Real-Time Communication
XML	Extensible Markup Language

“in the cloud, for the cloud” as they can easily provide an ample pool of compute resources for code development and testing, and code repositories to support developer collaboration, a key driving force to software productivity [4].

Traditional software development employs a tool chain including a text editor, a compiler and possibly a debugger, and performance analyzer. To accelerate software development, this rather disjoint tool chain is incorporated in an Integrated Development Environment (IDE) [5]. In order to further reduce time to market and development costs, software engineers heavily encourage reuse of existing software components in order to create new services.

To build a successful cloud programming environment, the advantages and functionality of traditional, desktop-based IDEs need to be maintained and augmented with additional features and strengths. Powerful code editors with a rich functionalities set (e.g. highlighting, autofill, etc.) are incorporated in web browsers. Compilation and testing execution are performed on cloud infrastructures and in several cases cloud providers can support deployment. Clear advantages of cloud-based IDEs include: (a) the access to wide pool of programming tools that are maintained by the provider, thus relieving the developer from the burden to setup, configure and upgrade their programming environments, (b) the ability to develop software without the use of powerful local computers, since the frequently compute-intensive tasks of the compilation and testing are performed elsewhere, and (c) the straightforward way to reuse code developed by other software engineers that share the same cloud environment.

In addition, Model Driven Development (MDD) can be valuable for cloud development environments. MDD is a SW development paradigm, where the analysis and development cycles are based on the creation and evolution of code or data models. Models are not just theoretical representations of entities, but primary artifacts of development through which developers design, implement and

interact with actual behavior of their software [6]. The benefit is a higher level of abstraction, which is required in the cloud in order to be scalable and agile. Also, the independence that MDD offers is considered a major advantage of a Model Driven Architecture (MDA), when designing a cloud based information system.

In this paper we present CIRANO, a cloud IDE that substantially supports MDD to facilitate the development of cloud applications. Beyond common functionality existing in modern IDEs like enhanced editors, debugging tools and support for popular databases, CIRANO supports (a) component-based application development, (b) multilayer programming, which greatly facilitates the online collaboration between developers, (c) automatic database development and connectivity and (d) an internal model and functionality repository. The paper presents the implementation and the technical details of the proposed CIRANO platform, along with an initial evaluation and comparison with existing systems. The paper is an extension of [7], which was presented at the Cloud Forward 2015 conference and contains a more detailed description of the CIRANO technical details along with a deeper evolution of the platform in practice.

The rest of the paper is organized as follows: In Section 2, we provide an overview of the existing Cloud Based Development Solutions. In Section 3, we describe the implementation of the proposed CIRANO platform, while in Section 4, we discuss its operation in practice presenting two case studies: development of an application from scratch and porting of an existing application. Finally, Section 5 concludes the paper.

2. Related work and background information

Cloud based development refers to the replacement of the developer’s classic desktop workspace with a set of cloud tools and components, which are accessed from any web browser at any time. Such a centralized workspace can be easily shared with other development teams or independent programmers, facilitating co-development. Any modern cloud based platform should include the following basic tools: a programming IDE, a local build system, a runtime environment, a database, some models, connections among them and descriptions of their dependencies. However, as developers use a wide range of programming languages and external utilities, it is impossible to gather everything in the cloud. These can be categorized as follows:

Cloud programming environments are online web based platforms designed to offer development capabilities, replacing the existing desktop default source code editors, pre-compilers, compilers, interpreters and linkers. They offer a wide range of programming languages in a browser environment which resembles a classic desktop development use-case. Apart from the default features and due to high competition, programming environments have evolved offering collaboration features like simultaneous source editing, connectivity with the most well-known repositories, virtual machines as runtime containers, server provisioning and selections of virtual machines with pre-installed services like databases and application servers [8–10].

Cloud repositories are web hosting services that leverage the strengths of well-known version control systems such as Git, Subversion and Mercurial. They are distinguished by the extension they offer, such as bug tracking, release management, integration with third party software, collaboration features and wiki-based documentation.

Cloud SW modeling tools refer to applications used to describe the functional and non-functional requirements of a project. Usually, they include a variety of design tools for graph creation and process representation. Their target is to imprint the architecture of an information system, a process and an interface or a component.

Download English Version:

<https://daneshyari.com/en/article/4950266>

Download Persian Version:

<https://daneshyari.com/article/4950266>

[Daneshyari.com](https://daneshyari.com)