CLOUD COMPUTING AND INTERNET OF THINGS FOR SMART CITY DEPLOYMENTS

GEORGE SUCIU[*], ALEXANDRU VULPE[**], GYORGY TODORAN[***], JANNA CROPOTOVA[****], VICTOR SUCIU[*****]

Abstract
Cloud Computing represents the new method of delivering hardware and software resources to the users, Internet of Things (IoT) is currently one of the most popular ICT paradigms. Both concepts can have a major impact on how we build smart or/and smarter cities. Cloud computing represents the delivery of hardware and software resources on-demand over the Internet as a Service. At the same time, the IoT concept envisions a new generation of devices (sensors, both virtual and physical) that are connected to the Internet and provide different services for value-added applications.
In this paper we present our view on how to deploy Cloud computing and IoT for smart or/and smarter cities. We demonstrate that data gathered from heterogeneous and distributed IoT devices can be automatically managed, handled and reused with decentralized cloud services.

Keywords: *cloud computing, open source, internet of things, smart cities, decentralized cloud*

Introduction

The paper addresses the convergent domain of Cloud computing and Internet of Things (IoT), that are nowadays two of the most prominent and popular ICT paradigms, together with the emerging topics of Big Data and Energy Efficient IT. The Cloud computing paradigm[1] is an umbrella term for computing services that are accessible over the Internet and developed on a common pool of remotely hosted resources.

There are different kinds of cloud computing services depending on the type of resources delivered via them, current state-of-the-art Cloud services can be such as Infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS), Network as a Service (NaaS), Storage as a service (STaaS), Sensor as a Service (SSaas) and others. These services permit to the users to obtain high reliability, increased security, high availability and to deliver Quality of Service (QoS) at low total cost of ownership (TCO). There are projects like RESERVOIR, VISION-CLOUD, OPTIMIS, CONTRAIL, that are cofounded by the European Commission, thru which were developed pan-European cloud services and also created new Cloud technologies like resource management, security or others and also relevant applications.

Correspondingly, the IoT paradigm[2] relies on the integration of a large number of heterogeneous devices, which are connected to the internet via different networking protocols. IoT

---

[*] PhD Student, Dipl. Eng, University POLITEHNICA of Bucharest, Faculty of Electronics, Telecommunications and Information Technology, 3CPS Department (george@beia.ro).

[**] PhD Student, Dipl. Eng, University POLITEHNICA of Bucharest, Faculty of Electronics, Telecommunications and Information Technology, 3CPS Department (alex.vulpe@radio.pub.ro).

[***] PhD Student, Dipl. Eng, University POLITEHNICA of Bucharest, Faculty of Electronics, Telecommunications and Information Technology, TEF Department (todoran.gyorgy@gmail.com).

[****] PhD Student, Practical Scientific Institute of Horticulture and Food Technology of Chisinau (jcropotova@gmail.com).

[*****] PhD Student, Dipl. Eng, University POLITEHNICA of Bucharest, Faculty of Electronics, Telecommunications and Information Technology (victor.suciu@beia.ro).

[1] P. McFedries, "The cloud is the computer," *IEEE Spectrum Online*, August 2008. Electronic Magazine, available at http://www.spectrum.ieee.org/aug08/6490.

[2] O. Vermesan and P. Friess "Internet of things-Global Technological and Societal Trends", Denmark: River Publishers, 2011.

permits the communication between different sensors connected to the Internet and the use of their services towards relevant applications. The concept was originally coined and introduced by MIT, Auto-ID Center and closely linked to RFID (Radio Frequency Identification) and electronic product code (EPC). Similarly the EC has financed a number of FP7 projects in the area of IoT (IOT-A, iCore, BUTLER, SENSEI).

The importance of our research derives from the fact that we propose ways thru which the Cloud and IoT concepts can be used in the context of Smart Cities. Therefore, our main objective is to propose a framework for cloud-based management of data received from sensor devices that can be used for Smart Cities scenarios. We will describe related work and their shortcomings that our research aims to address and present the adopted research methodology and implementation framework.

1.RELATED WORK

Already at the emerging of Cloud and IoT technologies, it has become clear that there is a need of convergence. Furthermore, integration with green energy efforts is a current hot research topic. IoT applications are using a wide range of heterogeneous distributed sensor network. They need to handle a huge number of sensor streams, and could benefit from the distributed elastic storage capacities of cloud computing systems. This cloud structures can boost the computational capabilities of IoT applications, permitting to several multi-sensor applications to perform complex big data processing that is subject to different QoS constraints. Other IoT services like large scale sensing experiments or smart city applications can benefit from a utility-based model, which is based on the on-demand delivery of IoT services over a cloud infrastructure.

A current state of the art about EU FIRE[3] projects is presented in Fig. 1.

Fig. 1. List of current IoT and Cloud eco-system

[3] EU FIRE project - http://www.ict-fire.eu/home/fire-projects.html

A number of projects originated from Machine-to-Machine (M2M)[4] but the current state of art of M2M platforms is quite fragmented and there is not a single view toward an interoperable smart object world. The M2M commercial platforms are vertically focused on solving specific sector issues and are tightly integrated with applications. This approach, taken from the telemetry legacy applications, has created a bunch of sensor devices not interoperable with each other, with high boundaries and integration possible only at database or presentation layers, without possibility of efficiently applying big data mining approaches.

For many years, M2M deployments were based on proprietary/custom applications and networking infrastructures, which were typically expensive to build, manage and maintain. Today's market for sensor devices is a hotbed of idea generation, as the prospect of embedding intelligence in the form of M2M technology into mobile devices has everyone excited about the possibilities[5]. The current market is already filled up with devices that can track everything from blood-glucose levels to traffic patterns.

As an example, SmartSantander proposes a unique in the world city-scale experimental research facility in support of typical applications and services for a smart city. The facility will comprise more than 20,000 sensors and will be based on a real life IoT deployment in an urban setting. The core of the facility will be located in the city of Santander, the capital of the region of Cantabria situated on the north coast of Spain, and its surroundings. SmartSantander will enable the Future Internet of Things to become a reality[6].

2. PROPOSED CONCEPT

2.1. Research Methodology

With our study we would extend the advances of IoT and cloud computing, by highly innovative and scalable service platforms through which to enable smart city services. The paper analyzes the gaps and designs solutions of how cloud computing is applicable to real-time data from embedded applications, thus unifying the IoT and cloud paradigms.

The vision is to propose a concept for a cloud open-platform integration of existing M2M approaches. The sensor and communication device manufacturing community is only one part of the model, where the application and service side is pushing for enablers to seamlessly access to widely distributed, real-time data from the environment. Throughout the recent past years, the concept of open-platform development, management and monitoring has emerged, basically solving again problems mainly from the technological point only in specific sectors.

In Cloud Computing, the word CLOUD is used as a metaphor for "the Internet," so the phrase cloud computing means "a type of Internet-based computing", where different services – such as servers, storage and applications – are delivered to an organization's computers and devices through the Internet. Similarly to IoT, a Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and

[4] M. Kranz, L. Roalter, and F. Michahelles, "Things That Twitter: Social Networks and the Internet of Things", *What can the Internet of Things do for the Citizen (CIoT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010)*, Helsinki, Finland, May 2010.

[5] G. C. Fox, S. Kamburugamuve, and R. Hartman, "Architecture and Measured Characteristics of a Cloud Based Internet of Things" *API Workshop 13-IoT Internet of Things, Machine to Machine and Smart Services Applications (IoT 2012) at The 2012 International Conference on Collaboration Technologies and Systems (CTS 2012)* May, 2012.

[6] Future Internet Research & Experimentation - http://www.smartsantander.eu/

presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers[7].

The concept will focus on a Smart City cloud interoperability and connectivity scenario based on the SaaS delivery model. This approach holds the immense promise of reducing capital and infrastructure costs while improving efficiencies of service provision within the Smart City framework. SaaS delivers software over the Internet, that eliminates the need to install and run the application on private servers, simplifies maintenance and enables customers to use applications remotely through IoT from anywhere in the world.

2.2. Conceptual framework

The framework proposes to design and implement an interoperable decentralized open source cloud platform for IoT applications, with the main objective to enhance existing M2M and their IoT foundations.

We will introduce in this chapter SlapOS, an open source Cloud Operating system which was inspired by recent research in Grid Computing and in particular by BonjourGrid[8]. Slapos is a meta Desktop Grid middleware for the coordination of multiple instances of Desktop Grid middleware. It is based on the motto that "everything is a process".

2.3. Cloud and IoT Architecture

SlapOS is based on a Master and Slave design. In this chapter we are going to provide an overview of SlapOS architecture and are going in particular to explain the role of Master node and Slave nodes, as well as the software components which they rely on to operate a distributed cloud for telemetry applications.

Slave nodes request to Master nodes which software they should install, which software they show run and report to Master node how much resources each running software has been using for a certain period of time. Master nodes keep track of available slave node capacity and available software. Master node also acts as a Web portal and Web service so that end users and software bots can request software instances which are instantiated and run on Slave nodes. Master nodes are stateful. Slave nodes are stateless. More precisely, all information required to rebuild a Slave node is stored in the Master node. This may include the URL of a backup service which keeps an online copy of data so that in case of failure of a Slave node, a replacement Slave node can be rebuilt with the same data[9].

It is thus very important to make sure that the state data present in Master node is well protected. This could be implemented by hosting Master node on a trusted IaaS infrastructure with redundant resource. Or - better - by hosting multiple Master nodes on many Slave nodes located in different regions of the world thanks to appropriate data redundancy heuristic. We are touching here the first reflexive nature of SlapOS. A SlapOS master is normally a running instance of SlapOS Master software instantiated on a collection of Slave nodes which, together, form a trusted hosting infrastructure. In other terms, SlapOS is self-hosted, as seen in Fig. 2.

---

7 R. Buyya, et.al. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities", *CoRR,* 2008.

8 J.P. Smets-Solanes, C. Cerin, and R. Courteaud, "SlapOS: A Multi-Purpose Distributed Cloud Operating System Based on an ERP Billing Model," *IEEE International Conference on Services Computing (SCC)*, pp.765-766, July 2011.

9 G. Suciu, C. Cernat, G. Todoran, V. Suciu, V. Poenaru, T. Militaru, and S. Halunga, "A solution for implementing resilience in open source Cloud platforms", *9th International Conference on Communications, COMM 2012, IEEE Communications Society,* pp. 335-338, June 2012.

SlapOS master nodes keep track of the identity of all parties which are involved in the process of requesting Cloud resources, accounting Cloud resources and billing Cloud resources. This includes end users (Person) and their company (Organisation). It includes suppliers of cloud resources as well as consumers of cloud resources. It also includes so-called computer partitions which may run a software robot to request Cloud resources without human intervention. It also includes Slave nodes which need to request to SlapOS master which resources should be allocated. SlapOS generated X509 certificates for each type of identity: X509 certificates for people who login, an X509 certificate for each server which contributes to the resources of SlapOS and an X509 for each running software instance which may need to request or notify SlapOS master. A SlapOS Master node with a single Slave node, a single user and 10 computer partitions will thus generate up to 12 X509 certificates: one for the slave, one for the user and 10 for computer partitions.

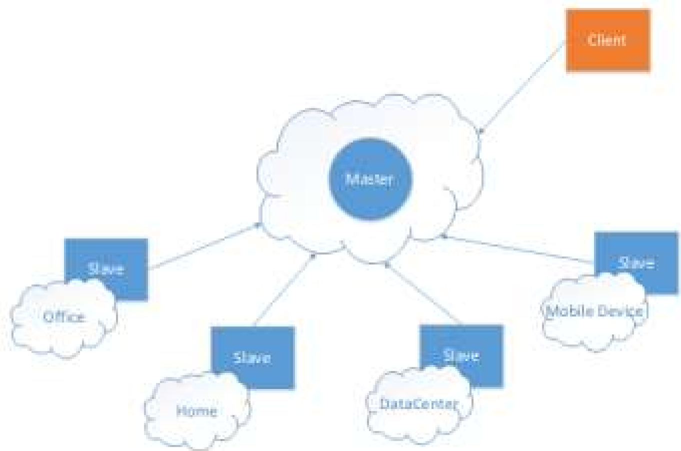

Fig. 2. SlapOS Master – Slave Architecture

Any user, software or slave node with an X509 certificate may request resources to SlapOS Master node. SlapOS Master node plays here the same role as the backoffice of a marketplace. Each allocation request is recorded in SlapOS Master node as if it were a resource trading contract in which a resource consumer requests a given resource under certain conditions. The resource can be a NoSQL storage, a virtual machine, an ERP with web-portal interface for displaying sensor data and Google Maps integration for RTUs localization, a Wiki, etc. The conditions can include price, region (ex. China) or specific hardware (ex. 64 bit CPU). Conditions are somehow called Service Level Agreements (SLA) in other architectures but they are considered here rather as trading specifications than guarantees[10]. It is even possible to specify a given computer rather than relying on the automated marketplace logic of SlapOS Master.

By default, SlapOS Master acts as an automatic marketplace. Requests are processed by trying to find a Slave node which meets all conditions which were specified. SlapOS thus needs to know which resources are available at a given time, at which price and under which characteristics. Last, SlapOS Master also needs to know which software can be installed on which Slave node and under which conditions.

[10] A. Vulpe, S. Obreja, O. Fratu "Interoperability procedures between access technologies using IEEE 802.21" *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology (Wireless VITAE)*, 2011, pp. 1-5.

SlapOS Slave nodes are relatively simple compared to the Master node. Every slave node needs to run software requested by the Master node. It is thus on the Slave nodes that software is installed. To save disk space, Slave nodes only install the software which they really need.

Each slave node is divided into a certain number of so-called computer partitions. One may view a computer partition as a lightweight secure container, based on Unix users and directories rather than on virtualization. A typical barebone PC can easily provide 100 computer partitions and can thus run 100 RTU web portals or 100 sensors monitoring sites, each of which with its own independent database. A larger server can contain 200 to 500 computer partitions.

SlapOS approach of computer partitions was designed to reduce costs drastically compared to approaches based on a disk images and virtualization. As presented in Fig. 3, it does not prevent from running virtualization software inside a computer partition, which makes SlapOS at the same time cost efficient and compatible with legacy software.

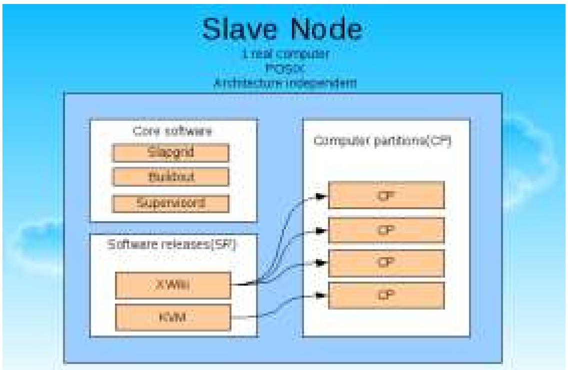

Fig. 3. SlapOS Slave Node

SlapOS Slave software consists of a POSIX operating system, SlapGRID, supervisord and buildout[11]. SlapOS is designed to run on any operating system which supports GNU's glibc and supervisord. Such operating systems include for example GNU/Linux, FreeBSD, MacOS/X, Solaris, AIX, etc

SlapOS relies on mature software: buildout and supervisord. Both software are controlled by SLAPGrid, the only original software of SlapOS. SLAPGrid acts as a glue between SlapOS Master node (ERP5) and both buildout and supervisord, as shown in Fig. 4. SLAPGrid requests to SlapOS Master Node which software should be installed and executed. SLAPGrid uses buildout to install software and supervisord to start and stop software processes. SLAPGrid also collects accounting data produced by each running software and sends it back to SlapOS Master.

Supervisord is a process control daemon. It can be used to programmatically start and stop processes with different users, handle their output, their log files, their errors, etc. It is a kind of much improved init.d which can be remotely controlled. Supervisord is lightweight and old enough to be really mature (ie. no memory leaks).

Buildout is a Python-based build system for creating, assembling and deploying applications from multiple parts, some of which may be non-Python-based. Buildout can be used to build C, C++, ruby, java, perl, etc. software on Linux, MacOS, Windows, etc. Buildout can either build applications

[11] J.P. Smets-Solanes, C. Cerin, and R. Courteaud, "SlapOS: A Multi-Purpose Distributed Cloud Operating System Based on an ERP Billing Model," *IEEE International Conference on Services Computing (SCC)*, pp.765-766, July 2011.

by downloading their source code from source repositories (subversion, git, mercurial, etc.) or by downloading binaries from package repositories (rpm, deb, eggs, gems, war, etc.). Buildout excels in particular at building applications in a way which is operating system agnostic and to automate application configuration process in a reproducible way.

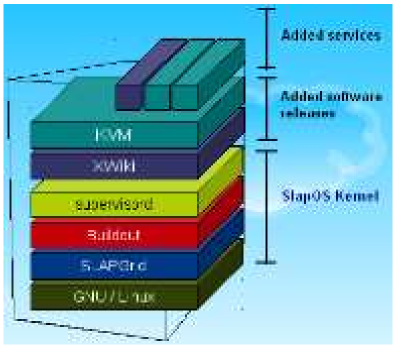

Fig. 4. SlapOS Kernel and User Software

Every computer partition consists of a dedicated IPv6 address, a dedicated local IPv4 address, a dedicated tap interface (slaptapN), a dedicated user (slapuserN) and a dedicated directory (/srv/slapgrid/slappartN). Optionally, a dedicated block device and routable IPv4 address can be defined.

CONCLUDING REMARKS

A Smart City deployment requires the integration of a distributed open sensor network and a decentralized cloud-based platform.

The current research explored the characteristics of a cloud platform for smart cities deployment with the focus on validating the platform's ability to deliver tailored IoT functionalities via the cloud middleware.

Furthermore, we will develop the proposed framework to include also applications for smart villages that will result in novel approaches for the monitoring of the agri-food supply chain and environmental monitoring in rural areas.

REFERENCES

- P. McFedries, "The cloud is the computer," *IEEE Spectrum Online*, August 2008. Electronic Magazine, available at http://www.spectrum.ieee.org/aug08/6490.
- Vermesan and P. Friess "Internet of things-Global Technological and Societal Trends", Denmark: River Publishers, 2011.
- EU FIRE project - http://www.ict-fire.eu/home/fire-projects.html.
- M. Kranz, L. Roalter, and F. Michahelles, "Things That Twitter: Social Networks and the Internet of Things", *What can the Internet of Things do for the Citizen (CIoT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010)*, Helsinki, Finland, May 2010.
- G. C. Fox, S. Kamburugamuve, and R. Hartman "Architecture and Measured Characteristics of a Cloud Based Internet of Things", *API Workshop 13-IoT Internet of Things, Machine to Machine and Smart*

Services Applications (IoT 2012) at The 2012 International Conference on Collaboration Technologies and Systems (CTS 2012) May, 2012.

- Future Internet Research & Experimentation - http://www.smartsantander.eu/
- R. Buyya, et.al. “Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities” *CoRR*, 2008.
- J.P. Smets-Solanes, C. Cerin, and R. Courteaud, "SlapOS: A Multi-Purpose Distributed Cloud Operating System Based on an ERP Billing Model," *IEEE International Conference on Services Computing (SCC)*, pp.765-766, July 2011.
- G. Suciu, C. Cernat, G. Todoran, V. Suciu, V. Poenaru, T. Militaru, and S. Halunga, “A solution for implementing resilience in open source Cloud platforms” *9th International Conference on Communications, COMM 2012, IEEE Communications Society*, pp. 335-338, June 2012.
- A. Vulpe, S. Obreja, O. Fratu ”Interoperability procedures between access technologies using IEEE 802.21" *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology (Wireless VITAE)*, pp. 1-5, 2011.