Cyber-physical systems clouds: A survey

Rihab Chaâri a,b,⁎, Fatma Ellouze b,c, Anis Koubâa b,d,f, Basit Qureshi d, Nuno Pereira f, Habib Youssef e, Eduardo Tovar f
a University of Mannouba, National School of Computer Science and Engineering (ENSI), Tunisia
b Cooperative Intelligent Networked Systems (COINS) Research Group, Raudh, Saudi Arabia
c University of Sfax, RedCAD Laboratory, B.P. 1173, Sfax, Tunisia
d Prince Sultan University, Saudi Arabia
e University of Sousse, PRINCE Research Unit, Sousse, Tunisia
f CISTER/INESC-TEC and ISEP, Polytechnic Institute of Porto, Portugal

A R T I C L E  I N F O

Article history:
Received 9 September 2015
Revised 12 August 2016
Accepted 17 August 2016
Available online 1 September 2016

Keywords:
Cloud computing
Cloud robotics
Cloud sensors
Vehicular cloud networks

A B S T R A C T

Cyber-Physical Systems (CPSs) represent systems where computations are tightly coupled with the physical world, meaning that physical data is the core component that drives computation. Industrial automation systems, wireless sensor networks, mobile robots and vehicular networks are just a sample of cyber-physical systems. Typically, CPSs have limited computation and storage capabilities due to their tiny size and being embedded into larger systems. With the emergence of cloud computing and the Internet-of-Things (IoT), there are several new opportunities for these CPSs to extend their capabilities by taking advantage of the cloud resources in different ways. In this survey paper, we present an overview of research efforts on the integration of cyber-physical systems with cloud computing and categorize them into three areas: (1) remote brain, (2) big data manipulation, (3) and virtualization. In particular, we focus on three major CPSs namely mobile robots, wireless sensor networks and vehicular networks.

As a new form of distributed systems for large-scale data processing, Cloud computing has been quite successful in integrating several applications areas including on-demand data centers [1], remote processing and storage [2], big data analytics [3], and resource virtualization [4], offering different categories of services to end-users. Basically, the emergence of Cloud computing comes as a result of the significant evolution of high-speed networks, coupled with the wide-spread usage of Service-Oriented Architectures (SOA) that allows providing any kind of resources as a “service” for machine-to-machine (M2M) interaction. In the Internet world, web services represent the major implementation of SOA with different and complementary approaches including Simple Object Access Protocol (SOAP) and Representational State Transfer (REST).

In the last four years, there has been an increasing interest for integrating the Cloud facilities in cyber-physical systems [5], which has led to the emergence of new fields such as Cloud Robotics [6–8], Sensor Clouds [9–11], and Vehicular Clouds [12,13]. Recently, a number of projects involving CPS have been conducted in areas such as agriculture, food processing, industry, environmental monitoring, security surveillance, and others. Meanwhile the number of publications related to cloud CPS has been growing rapidly. The authors conducted an extensive literature review by analyzing relevant research papers in order to help researchers understand the current status and future opportunities related to research in...
cloud CPS. This effort is focused on identifying the diversity of existing applications as well as proposed frameworks on cloud CPS in industrial systems as well as smart home environment utilizing industrial and/or service CPSs. Fig. 1 shows the trend of increase in number of publications listed in research databases including IEEE explore and science direct. The data collected from these databases highlights the number of journal as well as conference publications from years 2011–2015.

There are three main reasons for integrating CPS with the cloud. First, CPS devices, such as embedded sensor nodes or mobile robots, are typically resource-constrained devices with limited onboard processing and storage capacities. Even though some CPSs might have appropriate resources, it induces a much higher cost and makes them unaffordable for large-scale uses. This results in the need for offloading intensive computations from the CPS devices to much more powerful machines in the cloud, where very high computing capacity is available at low cost. Second, the amount of data generated and used by CPSs is inherently very large as CPS devices are in continuous and direct interaction with the physical world getting instantaneous data to feed the computing system. Analyzing and extracting useful information from such large volumes of data require powerful computing resources. In addition to the need of offloading computation, abundant storage resources becomes a major requirement to cope with big data manipulation. Third, cyber-physical systems are heterogeneous in nature, making interoperability a serious challenge. Accessing these heterogeneous systems from the Internet is not straightforward without having standard and common interfaces. To overcome this issue, there is a need to virtualize their resources and expose them as services to facilitate their integration.

Considering the aforementioned challenges, it is clear that CPSs have three main requirements namely, (i) offloading intensive computation, (ii) storing and analyzing large amount of data, (iii) enabling seamless access through virtual interfaces. These are the major reasons why cloud computing represents a promising solution that addresses all these three challenges because, if coupled with a high-bandwidth connection to the CPS, it provides storage and computing resources as services to the back-end CPS, and allows for easier integration of these CPSs by virtualizing the access to them through standard web service interfaces. Fig. 2 explains how cloud computing can help to make cyber-physical systems accessible and more powerful.

In this survey paper, we first give a motivation on how cloud computing can enhance CPSs. Next, we give in Section 3 an overview on cloud computing. Then, we present a comprehensive overview of latest research works on the integration of cloud computing into three CPS areas namely, robotics (Section 5), wireless sensor networks (Section 5) and vehicular networks (Section 7). The integration is categorized along three axes: (1) computation offloading (or remote brain), (2) big data analytics, and (3) virtualization.

2. Motivation

The promising application of cyber-physical systems raised new functional requirements including real-time processing, storage, and accessibility and non functional requirements like security. In this research work, we are interested in the functional requirements.

2.1. Real-time processing

With the recent technological developments, cyber-physical systems can provide finished products and high quality services. This may pose the problem of executing many useful programs. For instance, a mobile nursing robot [14] should autonomously achieve the quality of care that nursing staff can provide. From a medical informatics point of view, this kind of robot should read environmental data, perform intelligent navigation which may include object detection and/or tracking, support a robust and safe human-robot interaction, etc.

While moving on the road, vehicles are also offering a certain number of applications and services simultaneously to achieve human safety and satisfaction [15]. Offering an acceptable level of safety requires the execution of cooperative collision warning, incident management, and emergency video streaming. Besides, vehicles should execute a set of basic applications like platooning, vehicle tracking, and notification services. Drivers also may look for some comfort applications like parking place management, peer to peer applications, and entertainment.

Therefore, cyber-physical systems should be capable of executing a variety of applications completely independently. Unfortunately, the design of these systems usually imposes stringent constraints on computing performance. The limited resources, such as the memory sizes, the battery capacity, and the processor speed, cannot satisfy the demands for executing this variety of complex applications.

Many research works have faced the problem of real-time processing in cyber-physical systems. As a matter of fact, authors in
[16] designed and developed an IoT cloud platform to connect smart devices (e.g. sensor nodes, robots) to cloud services for real-time processing of data and control. In particular, they considered the deployment of robotics applications with strict processing delay and scalability requirements. They were motivated by the fact, that for some resource constrained devices like sensors and robots, only a small portion of data can be processed locally and the remainder should be offloaded to a central server. A Roomba vacuum cleaner robot was provided as an example. Experimental studies involved a Turtlebot robot with a line follower application and performance evaluation study demonstrated that it is possible to achieve low delays with commodity hardware in the cloud and it is possible to scale the architecture to hundreds of connected devices.

Another example that needs real-time processing in sensor networks and Internet of Things is IoT analytics. In [17], the authors introduced a new approach to IoT data stream analytics for real-time data acquisition, annotation and processing of sensor data. The paper uses the OpenIoT platform of an EU FP7 project to develop a framework for offloading data from sensor networks for real-time data analytics in IoT applications. The server deploys complex clustering algorithms to analyze data in real-time. Such a tool is valuable for understanding complex phenomena such as impact of air pollution in human health.

2.2. Storage

The rapid growth of the Internet of things with the latest advances of Information Technology (IT) have increased the amount of data generated from cyber-physical systems. These systems are now all over the world collecting data. Smarter vehicle management, home automation systems, surveillance, etc are all applications where cyber-physical systems generate a massive amount of data that should be stored somewhere for analysis.

The exponential growth in the amount of data generated by cyber-physical systems needs a revolutionary storage strategy. Memory or/and storage systems are fundamental performance metrics and energy bottleneck in these systems [18,19], in addition to communication. In the design of cyber-physical systems memory issues play a key role, and significantly impact their performance. This memory capacity must scale with applications requirements in terms of storage. Unfortunately, this scaling exacerbate the weakness of storage capabilities in these systems. This represents an additional reason for the need to offload data and computation to the cloud.

2.3. Accessibility

The confluence of advances in wireless communication has led to the mobility of cyber-physical systems. They became deployed in applications where mobility is a necessity. For example, vehicular networks are highly mobile networks as vehicles are consistently moving, which induces additional challenges with respect to protocol design and performance issues. In [20], the authors discussed the state of the art of IP mobility support solutions for vehicular networks to ensure their accessibility considering the two major kinds of mobile communication models, namely the vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I) communication.

The major challenge with mobile CPSs is that they might be inaccessible either permanently or intermittently if mobility support is not carefully addressed. Handoff and handover mechanisms are typically integrated in mobile network communication protocols to deal with mobility. In [21], the authors proposed a handoff mechanism for low-power wireless sensor networks to maintain the accessibility of sensors and evaluated its performance through a probabilistic model. The paper concludes that the handoff mechanism performs well in the transitional region of the link quality [22], and may achieve up to 98% of throughput with delays in the order of tenths of seconds. The use of cloud computing would help improve the effectiveness of these applications with mobility requirements. As an example, in [23], the authors proposed a
mobility-aware trust worthy crowdsourcing framework in a cloud-centric IoT architecture. The framework incorporates user mobility awareness while considering malicious alteration of data. They came to the conclusion that mobility-awareness and reputation-awareness in crowdsourcing IoT application improve the effectiveness of smart city management authority.

Most of the limitations identified above will be remedied with cloud computing. In the following sections, we will discuss how researchers addressed these challenges.

3. Cloud computing in a nutshell

According to the National Institute of Standards and Technology (NIST) [24], cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. A simpler definition of cloud computing is having access to all your applications and data from any network-connected device. This approach is not entirely new since the concept of delivering computing resources through a global network was introduced in the sixties by John McCarthy who proposed the idea of computing being delivered as a public utility [25]. With the evolution of Web 2.0, cloud computing has evolved through several phases including grid and utility computing until the delivery of the first Software as a Service (SaaS) with Amazon EC2/S3.

The service model in cloud computing defines the type of services you can access in the cloud. Three service models are universally accepted: (1) Infrastructure as a Service (IaaS), where the delivery of hardware infrastructure over the network on pay by use model, (2) Platform as a Service (PaaS), which provides a software environment where consumers can deploy their applications, (3) Software as a Service (SaaS), which offers a complete operating environment that contains management, applications, and the user interface.

These services can be deployed in four models: (1) Public Clouds: cloud services in public clouds are made available by a service provider for open use by the general public. It is owned by a third-party selling cloud services like Amazon AWS [26], which owns the infrastructure and offers the access via Internet. (2) Private Clouds: are built for exclusive use of single client/organization. It may be managed by the organization itself or a third party. Despite being an expensive solution, the private cloud basically offers the highest level of security and control. (3) Hybrid Clouds: represent a combination of two or more distinct clouds to accomplish various functionalities within the same organization. For instance a collaboration between VMware [27] and Google cloud [28] has offered an hybrid cloud solution called VMware vCloud Air. (4) Community Clouds: represent a collaboration between several organizations to serve a common function or purpose. It may be operated, owned, and managed by a third party or the constituent organization(s). A good example of community cloud deployment is Apps.gov [29] which provides cloud services to federal agencies.

The definition of cloud computing requires that the practical design of cloud system must consider the following five keys characteristics:

- Scalability: the cloud needs to dynamically increase the resources to serve the increasing workload as and when required. Scalability in cloud computing means: (1) the ability of an application/product to function well even when it increases in volume or size to meet the user requirement, and (2) taking full advantage of the rescaled situation.
- Usability: cloud computing services are supposed to serve users from different domains. To achieve clients satisfaction, the access to cloud should be as simple as possible by designing a comprehensive and a user-friendly interface to cloud.
- Reliability: it refers to the probability of a system, including all of its hardware and software components, to perform correctly as expected. With third dependencies on remote services and databases, network connection and hardware cloud services are downright complicated. As such, reliability of the cloud is a major concern considering its distributed and multi-tenancy nature. In [30], the authors discussed security and reliability issues in cloud computing systems. A cloud system should provide the services it is designed for with no major flaws and this induces additional challenges to their design and development. Verification, validation and testing techniques should be used to ensure reliability of these systems.
- Security: when you consider moving your application to the cloud, consumers must understand the risks and security associated with the cloud. They should ensure the privacy of their sensitive data [31]. Security and privacy presents a lot of challenges in today cloud computing systems, like identifying new threats and vulnerabilities, protecting virtual infrastructures, protecting outsources of computation and services, securing big data storage and protecting user data [32–34]. The use of integrated security solutions at different levels that span over authentication, encryption, intrusion detection and prevention systems, trust management, etc., should be carefully designed based on the application requirements.
- Cost: the cost is one of the most important factors regarding the use of cloud computing platforms. It is generally based on the concept pay-as-you-go taking advantages of the elastic computing capabilities provided by the cloud. However, deciding whether the cost of subscribing to cloud computing facilities or building its own data center is a highly complex problem that depends on several factors, as lower cost is not always guaranteed when using cloud services. A cost effectiveness analysis should be conducted when cloud computing platform is to be used [35].

4. Cloud robotics

Nowadays, we are at the cusp of a revolution in robotics. The current research trend aims at broadening the usage of service robot systems as well as industrial robots in various applications including smart home environments [36], smart office buildings, airports [37], shopping malls [38], manufacturing labs, integrated industrial control systems, etc. The demand for these service robots is increasing in a world with significant attachment to the use of Information and Communication Technologies (ICT). The International Federation of Robotics reported that the number of service robots for personal and domestic use sold in 2014 increased by 11.5% compared to 2013, augmenting sales up to USD 3.77 billion [39]. It is also forecasted that the sale of household robots could reach 25.2 million units during next three years.

Although service robots have been useful for several applications and various uses, their use is limited comparatively to other technologies such as smartphones, mobile phones, tablets, etc. Essentially, robots have limited exposure to the general public at a large scale for the following reasons:

- Robots have been typically used as standalone systems for very specific and dedicated missions in controlled environments, such as in industrial manufacturing [40], hospitals [41] etc. Even in the case of cooperative and multi-robots applications, robots communicate with users or other robots to perform specific pre-programmed, pre-defined missions, yet are isolated from the external world. Consequently, they cannot learn from their context.
• The complexity of configuration and maintenance of the robots make their use challenging for non-technical and non-computer-savvy users.
• The relatively high cost of sophisticated service robots can be cost abhorrent for public users which confined their utilization in professional organizations or specialized domestic applications.

Given the presence of the aforementioned restrictions, there is an increasing demand and interest in using the Internet infrastructure as a means of promoting robotics applications from two perspectives; by exposing robot resources as Internet services to end-users; and by enabling the robot-to-robot (R2R) as well as robots-to-end users communications through the Internet.

With the emergence of cloud computing, robots may also reap benefits from the huge computing resources available in the Internet to increase their processing capabilities for computation intensive applications. The cloud robotics paradigm, coined in 2010 by James Kuffner, seems to be the missing building block towards jumping to a new frontier of public use of robots. Indeed, cloud robotics has opened several new research ideas and interest with respect to coupling the Internet and cloud computing with robotics.

Cloud computing leverages the limitations of embedded hardware by enabling the robots to offload computation and massive storage requirements to the remote cloud infrastructure which would be used as a “remote brain”, where all the computation would be done based on the sensor data accumulated from the robot. Furthermore, advances in networking technologies and high speed mobile networks have enabled fluid interaction between the cloud and the robots. Benefits of aforementioned enabling technologies have unveiled the possibility of sharing and collaboration among robots, machines, smart objects and humans.

Cloud robotics evolved over the last few years from the concept of networked robotics. With the availability of the world wide web, online web robotics became a reality. In the nineties, some of the earliest work by K. Goldberg et al. [42] [43] resulted in web based tele-operated robots controlled remotely via an Internet browser. In [44], McKee described the benefits of distributed processing in networked robotics and challenges in communication, synchronization and performance. In 2010, Kuffner [45] coined the term, “Cloud Robotics” in his paper on cloud-enabled robotics, providing a wide vision where robots can harness the massively parallel computation resources and vast data storage available via the Internet.

In this section, we will present three facets of cloud robotics: (1) Remote Brain: Offloading Computation, (2) Remote Storage: Big Data Processing, and (3) Remote and Distributed Collaboration.

4.1. Offloading computation

Nowadays, applications requiring massive computations can be remotely accomplished in the massively parallel cloud infrastructure. It has become possible for mobile and service robots to overcome the limited onboard resources by “offloading” massive computation tasks on remote servers via the Internet. Consequently, in the near future, robots would be used as actuating devices, capturing and offloading sensor data to the cloud infrastructure for rapid processing and analysis. Upon completion of the processed tasks, robots would actuate accordingly, driven by the response received from the cloud.

Robots are typically capable of processing information using multiple onboard processors. However, certain types of robots such as mobile robots or Unmanned Aerial Vehicles (UAV) may have limited on-board resources lending to stringent dimensional size and limited payload requirements. As an example, surveillance robots [46] require performing real time mapping, localization, autonomous navigation and object recognition. The robot can capture sensory data from its environment and send it to a cloud service. The cloud service permits running interactive, real-time processing tasks in parallel, including applications for video surveillance, image building using Point Cloud libraries (PCL), sensor information tracking, predicting and evaluating performance and other tasks.

Recently, researchers have addressed the concept of “offloading” from robots to cloud infrastructure. Rapyuta proposed by Hunziker et al. in [47] is the ROBOEARTH cloud engine devised to overcome the challenge of managing robotic resources constraints by implementing an open source cloud robotics framework. It uses Amazon data center to allow outsourcing of robots on-board computational processes by providing secure, elastic, customizable, and ROS compatible computing environment in the cloud. This ground breaking work presents opportunities for a Software as a Service model that may allow custom applications to offload data from robots to the cloud service, which can process the data and return outputs. This allows the users to avoid the burden of processing and managing data onboard the device. In [48] L. Bingwei et al. present Cloud Enabled Robotics System (CERS) that allows robots to mesh together to form a local networked system. These robots can offload heavy computation tasks to the server side implementation of a cloud enabled software architecture. The proposed system also considers security implications while merging cloud and robot networks. Researchers tested the system for an application of real-time video tracking and provided performance comparison based on virtual machines and physical machine clusters. Another example of computation offloading is presented in [49] where a cloud robotic system with real time face recognition is implemented. Due to the resource-hungry face recognition applications requiring complex computations from resource-constrained robotic systems, researchers exploited the computational services of the cloud by offloading heavy computational tasks from robots to cloud servers.

Turnbull in [50] developed a full scale cloud infrastructure to control the behavior and formation of a multi-robot system by offloading computation tasks to the cloud. Robots within the system are designed with minimal hardware, equipped with a microcontroller with WiFi capabilities for robot-to-robot (R2R) as well as robot-to-cloud (R2C) communications. All the computational workload and execution of algorithms required to control the robot behavior are performed in the cloud. The cloud infrastructure receives data from a vision acquisition system, determines the robot location and status, implements algorithms to control its behavior, and then sends appropriate commands to the robot. The system was validated using iRobot Create equipped with LinkSprite Diamondback microcontroller, a cloud infrastructure using XenServer as the hypervisor and XenCenter as the virtualization manager, and a single virtual machine with Ubuntu 12.10 and MATLAB installed.

Authors in [51] have exploited the Service Oriented Architecture (SOA) and cloud computing technologies to perform robotic applications like path planning, object recognition, map building and navigation in the cloud using the Map Reduce computing cluster. The designed framework is composed of three main components: (1) Cloud manager which receives incoming service requests via http/https and checks the authorization of requesting client, (2) Robotic service handler to execute and check the availability of requested robotic applications exposed as robotic services, and (3) Map Reduce computing cluster to process large amount of data. The authors have demonstrated the effectiveness of Map Reduce computing cluster by performing a real time speech based navigation of create robot using VPL of MRDS to simulate and control the robot. The authors discuss in detail the performance aspects of execution of map-reduce on the cluster. Much needs to be desired in terms of validation of the architecture under execution of
real-life applications scenarios. Another application of Map-reduce framework executing on Hadoop is the DAvinCi project reported in [52]. The researchers provide the DAvinCi server that acts as a proxy server coupling the two eco-systems, Hadoop distributed File System (HDFS) and Robot Operating Systems (ROS). The service provides a publish/subscribe model for robots to send/update their information as well as requests, which are logged and maintained on the server. The server pushes this data and/or requests to the backend HDFS. The server triggers MapReduce tasks to execute data and collects results produced by the Hadoop ecosystem.

Finally these results are transmitted to the ROS subscribers. They validated this work by implementing the FastSLAM algorithm and reported significant performance gains in execution times provided by the framework. Although the focus of the paper is the evaluation of FastSLAM algorithms under large distributed environments, the communication issues between internal processes as well as the cloud is not detailed. We notice that the implementation of the server component is dependent on reliable communications between the namenode in the hadooop cluster and the ROS master-node. Although hadoop provides backup measures by invoking secondary namenode in-case of the primary namenode failure, the proposed framework lacks to address the reliability and communication latency or failure issues of the DAvinCi server between the two ecosystems. The researchers implemented the system using a hadooop cluster composed of only eight nodes. They plan to address failsafe mechanisms for communications between ROS ecosystem and the DAvinCi server.

R2R and R2C Communications are of utmost importance for computation offloading applications. To address the shortcomings of the DAvinCi project, authors in [53], developed a system for real-time moving objects recognition and tracking using cloud computing to perform offloading of heavy tasks. To guarantee real-time performance, they developed an offloading decision framework which estimates the computation and communication times for tasks and decides to execute the tasks either on the robot or the cloud server. The resulting decision minimizes the execution time while maximizing tasks completion rate while satisfying real-time constraints. While the authors performed computation offloading, they have not used cloud computing platforms or technologies, but rather a simple server machine. In addition, the deployment is rather limited to a local area network with wireless connection, and not tested in Internet infrastructures. Furthermore, it is applied for a stationary robot, which limits its applicability to such context. Another example of research in reliable communications is presented in [49]. Researchers address the data transfer latency between the mobile robot and the cloud server in order to perform real time face recognition. They proposed an architecture that supports high-speed data transmission and distribution of computation load.

Several knowledge databases for robots have recently been proposed. Robo Brain [55] is a project created to build a shared, large-scale knowledge base for robots to be able to function in human environment. This project helps robots to learn human’s behavior. The core idea of this project is to enable newly deployed robots in an environment to connect and learn from the robo-brain knowledge-base instead of beginning from scratch. The Columbia Grasp dataset [56] is a large dataset of pre-computed grasps on thousands of 3D models designed for learning and benchmarking purposes. To construct this database, authors in [57] have combined (1) RoboEarth database knowledge to give a description of robot tasks with (2) the distributed task execution methods of the Ubiquitous Network Robot Platform [58] to abstract the access to robot hardware and software components. The Willow Garage Household Objects Database [59] contains thousands of 3D models used to evaluate various aspects of grasping.

Google Goggles, [60] a free image recognition service for mobile devices, has been incorporated into a cloud-based system for robot grasping. Users can take a photo of an object and send it to Google for analysis and storage. Authors in [61] have designed a knowledge base about robotics rehabilitation. The database information is made available through web-interface allowing robots designers to update information about their robots rehabilitation. REHABROBO-ONTO is presented in Web Ontology Language (WOL) based on the standards of World Wide Web Consortium (W3C) to enable integration with medical ontologies. The concept of shared knowledge database was also used with industrial robots to meet human’s intelligence when working in dynamic environments. The European Union’s ROBOEARTH project [62] was a leader of the remote memory idea. The main objective of this project is to build a World Wide Web for Robots for sharing knowledge about actions, objects, and environments between robots [63]. Robots can store and share information in a common presentation designed by the RoboEarth language.

Authors in [64] have designed a rehabilitation database using physical therapy robots. Therapy robots equipped with the required material can record and send the data to a rehabilitation server which in turn saves the data and maintains the database. Performance evaluation has proven that the results using the rehabilitation database are acceptable compared to those generated by therapists. However, these results remain theoretical as the database was not tested on real patients. Ben Kehoe et al. in [65] have designed an architecture in which they connect robots to the Google Goggles image recognition system. The robot sends data collected by its depth sensor and camera to a Google server that performs object recognition and returns a set of grasps with confidence values. A prototype of this architecture was developed and evaluation results showed comparable performance.

4.3. Virtualization

Industrial and personal robotic systems are designed to provide services to human beings in a variety of circumstances. Sets of robots accessed remotely can collaborate together to perform various functions, tasks and sub-tasks to achieve a unified goal. There are several issues and challenges that require attention: (1) robot programming and manipulation is usually complex for beginners and naive users, (2) monitoring and controlling robots usually rely on complex programming routines that cannot be handled by end-users, (3) robots are typically expensive platforms and are not affordable for a large audience. In this context, researchers profit from cloud computing concepts and recent advances in web technologies to virtualize robotic systems, hiding complexities and offering robot-as-a-service to the end-users.

Authors in [66] present Jeeves, a distributed service framework for cloud based robot services. The key mechanism of the proposed
framework is a robot assignment function, which discovers distributed robot resources and assigns the requested tasks by end users to suitable robots. Jeeves integrates various devices including robots with Internet services and provides a platform for offloading and executing computations in the cloud. Despite authors have proved the effectiveness of their architecture, they didn’t provide a secure access to the cloud.

Authors in [67] have designed and implemented a cloud infrastructure that virtualizes robots and delegates tasks to robots. When the system receives a robotic application to perform, it divides the task into sub-tasks and assigns the sub-tasks to local robots and robots that belong to other clouds. The authors defined a robot node level virtualization that allows robots to run multiple applications at the same time. The designed system comprises three main components: a P2P overlay, a robotic cloud and a VRC (Virtual Robotic Cloud). The P2P overlay is used as the interaction network between different robotic clouds through the VRC which is defined as a node that facilitates the communication between robotic clouds via the overlay. The architecture model presented in this work is a gateway-based model, therefore it is of high interoperability. However, the gateway is a failure point which may affect the architecture robustness.

The Unmanned Aerial Vehicles (UAVs) area also attracted recent research works on SOA and REST Web services. In [68] authors proposed a mapping between cloud computing resources and UAVs resources based on a SOA model. They offer two kinds of services: essential services (e.g. mission organization service, broker service, ground station commands, etc.) and customized services (e.g. sensing services, actuation services, image/data analysis, etc.). The paper only provides high-level description of system architecture, components and services without any specific detail on how these could be implemented on a real system. In addition, the authors did not discuss performance metrics and tradeoffs related to the use of UAVs in the cloud, and no quantitative performance analysis was presented. In addition, scalability issues with the deployment of multiple UAVs on the cloud were not discussed. In [69], the same authors improved their previous work and designed a RESTful web services model by following a Resource-Oriented Architecture (ROA) approach. They have also designed a broker that dispatches mission requests to available UAVs. The broker is responsible for managing the UAVs, their missions and their interactions with the client. A prototype for emulating the UAV and its resources was implemented on an Arduino board. This prototype is very limited as it does not demonstrate a sufficient proof of concept on real drones or robots which does not effectively demonstrate the feasibility of the approach.

In [70], Koubaa proposed RoboWeb, a SOAP-based service-oriented architecture that virtualizes robotic hardware and software resources and exposes them as services through the Web. RoboWeb consists of the integration of different Web services technologies with the ROS middleware to allow for different levels of abstraction (multi-layer architecture). It was designed to develop a remote lab which allows researchers and students to remotely access and monitor robots that belong to the framework. This system architecture comprises three main layers: (1) the web interface layer which implements rosPHP library that provides ROS functionalities allowing access to ROS-enabled robots, (2) the roboWeb service broker which is defined as a middleware allowing interaction between end-users and ROS-enabled robots, and (3) the robotic ecosystem which comprises ROS-enabled robots including Tutlebot and Wifibot. Researchers have validated their proposed architecture by developing a web interface that allows non technical users to access, control and monitor robots that belong to the framework. Unfortunately, the architecture offers a low level of security. Password-based authentication doesn’t adequately provide a strong security.

In [71], the authors proposed a remote robot lab for conducting shared experiments and developments that allows the control of robots remotely through a web interface. They made extensions to the Robot Operating System (ROS) middleware using rosjs, and rosbridge, through web sockets accessible anywhere over the Internet, and provides security mechanisms and runtime tools for remotely manipulating the robots. They also present two robotic experimental environments that implement rosjs technology in order to visualize and interact with complex robot platforms.

Rsi Research cloud proposed in [72] is a cloud computing platform that provides robotic services through the Internet. Authors exploited the Robot Service Network Protocol (RSNP) to hide robots complexities and expose robotic services to non-experts in robots. They have tested their proposed architecture by implementing a surveillance camera service.

Chen et al. in [73] defined the paradigm Robot as a Service (RaaS), proposing a cloud framework for interacting with robots in the area of service oriented computing. The authors exploited the SOA to design and implement a prototype of the Robot as a Service (RaaS) cloud computing model which consists of exposing robotic services and applications and allows interaction with developers and end-users. The design complies with the common service standards, development platforms, and execution infrastructure, following the Web 2.0 principles and participation. RaaS unit is composed of three main parts: (1) service provider allowing to add/remove services to/from the robot, (2) service consumer allowing end-users and developers to benefit from deployed services and compose new applications, and (3) service broker for research, discovery and publishing of robotic services and applications in the directory.

In Table 1, the architectures for integrating robots with cloud computing are compared with respect to the key requirements presented and discussed in Section 3.

5. Sensor clouds

WSNs [74] are considered as one of the enabling CPS technologies in the 21st century [75] and one of the major contributors to the IoT. WSNs currently represent an invincible technology embedded into a wide variety of industrial and CPS applications, including healthcare [76], environmental monitoring [77], defense and military [78], industrial automation [79], smart environments [80], smart grids [81], to name a few.

Over the last fifteen years of WSNs research, the major focus has been on the design of lower layer protocols including communication, medium access control, networking, topology control protocols and these reached a certain saturation. In addition, with the maturity of the field, the current trend nowadays is to adopt the use of standard communications and networking protocols rather than devising proprietary solutions. Very recently, and with the emergence of the IoT paradigm, the focus is switched to be more on the integration of WSNs into the Internet, which raises several new challenges, including compatibility and interoperability issues, and interfacing between the WSNs world and the Internet. The first integration step came with the design of the IPv6 version for low-power networks, namely the 6LoWPAN, which enabled the flow to run between wireless sensor devices and the IP World. Later, several other networking protocols emerged such as RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) [82] and LOA/DnG (6LoWPAN Ad Hoc On-Demand Distance Vector Routing) [83] routing protocols. The second step was the integration at the application layer by enabling IP network users to access sensor data through Web interfaces. SOA and Web Services [84] [85] played an important role into this integration, in particular the use of RESTful Web Services as they are lightweight and can be more easily sup-
reported by low-power devices such WSNs as compared the SOAP Web Services.

All these efforts have paved the way to a more consistent integration of WSNs with the Internet using the cloud paradigm and resulting in what is called Sensor Clouds. Considering the limited resources of typical wireless sensor nodes, their integration with the cloud is an important milestone to leverage WSN applications and promote their performance and capabilities. In fact, with the emergence of the Internet-of-Things (IoT), wireless sensors are embedded nowadays in trillions of devices around the globe and keep generating data in real-time from the surrounding environment.

Typically, sensor-enabled devices are merely data generators and are too restricted to perform extensive operations to analyze and make use of the data they generate. However, their wireless capabilities allow them to transfer their data to more powerful devices to make use of this data. It is here where the cloud comes into play by providing different computing and storage resources, in addition to a wide array of applications and utilities to process and manipulate the big data. Furthermore, users may need to access particular sensor devices to read data or modify its settings. For an effective user-sensor interaction, interfaces and management layers are required. The cloud provides such facilities through different technologies centered around service oriented architecture and Web services.

There has been several research efforts towards the design and deployment of sensor clouds [86] to offer a remote sensor data storage, management and integration. In what follows, we present the major contributions in the field, and discuss the opportunities offered by sensor clouds in the three mentioned perspectives (1) Offloading Computation, (2) Big Data, and (3) Virtualization.

5.1. Offloading computation

With the increasing popularity of WSN in various domains, a wide range of potential and promising applications is evolving and requires processing large amount of sensor data in real-time to cope with the needs of complex information processing algorithms such as object detection and tracking, sensor-based surveilance, control and automation, etc. These applications are quite demanding in terms of processing and storage and require the use of powerful and scalable processing infrastructures to provide the needed computation services to low-capability sensor devices.

The current research trend aims at harnessing cloud computing resources to augment resources capabilities of WSNs by migrating heavy processing applications to the cloud, which save energy and improve their performance.

A significant amount of research has been performed on offloading computation to the cloud. With the emergence of the IoT paradigm, ANGELS [87] was proposed as a framework to offload huge volume of data and to provide a parallel processing infrastructure among smart devices surrounding us, such as mobile phones. The key idea was to use mobile devices when they are idle to execute programs and process data, thus contributing to a new paradigm of distributed computing in the IoT. The cloud architecture consists of a server machine with a HADOOP MapReduce framework which mediates between end-devices for job offloading. The server module comprises three components: (1) the Data Partitioner, which partitions input data according to the score of devices calculated using AndEBench benchmarking tool for android devices, (2) the Job Scheduler, which maps the data partitions to the nodes, and (3) the Combiner which combines results received from mobile devices and produces the final result. The authors have experimentally validated their proposed framework by estimating the value of pi using four android mobile phones. Although still in an experimental phase, ANGELS framework provides a new insight on how low power devices like smart phones could be used behind a cloud for remote and distributed processing. However, the efficiency of such a system remains an open question considering that these mobile devices cannot handle extensive computations. We note that this approach might seem to contradict the general concept that computation should be offloaded from low-capability devices such as smartphone to a cloud infrastructure with higher performance commodity hardware. However, in general sense, it follows the same concept, as in the ANGELS context, computation can indeed be offloaded to smartphones, which might have low processing and storage capabilities, but their integration behind the cloud may provide sufficient resources for some kinds of applications like those considered in ANGELS.

Authors in [88] tackled the problem of the limited processing resources in WSNs. They proposed a component-based model architecture that integrates WSNs with IaaS Hybrid Cloud to allow for real-time data transmission and processing. The architecture is composed of three main tiers: (1) the sensor tier which uses a sink node to collect and send sensor data to a local proxy, (2) the gateway tier that consists of a local proxy that uses LooCI middleware to dynamically generate, register and control components for each detected sensor node. Then, the local proxy transmits generated node components to the cloud and send sensor data to the related component for processing via the Event Bus of the LooCI middleware, and (3) the cloud tier which consists of a hybrid cloud that combines Eucalyptus Private Cloud [89] and Amazon Web Services [26]. The authors experimentally proved that the proposed architecture induces only a very little overhead in terms of latency with an average of 0.6 ms for the invocation of dynamic components. In addition, it minimizes the memory footprint on sensor devices and reduces energy consumption.
In [90] the authors proposed a cloud-based mobile healthcare system, which enables end-users to control, monitor and improve their health using mobile devices. The limitations of the mobile devices are avoided by migrating computationally intensive operation to cloud servers. Two multi-cloud offloading approaches were proposed. The first model is a self-reliant architecture, which intends to promote stability at the cost of a higher communication overhead; while the second is called multi-cloud offloading, which provides a better tradeoff between stability and communication cost. However, the authors do not give any tangible implementation of such a system, and only provides an analytical study on the performance of both approaches.

In [91], the cloud was used to offload GPS signals to be processed by the cloud instead of being processed by the mobile device. The main motivation behind this was to reduce energy consumption of mobile phones. The key idea consists of exploiting cloud computing resources to generate a number of candidate landmarks, and then use other geographical constraints to discard the wrong solutions. It was experimentally proven that using the proposed Cloud-Offloaded GPS (CO-GPS) solution, sensing a GPS location takes 3 orders of magnitude less energy than GPS on mobile phones. While this solution was shown to be effective in terms of energy efficiency, it has some limitations with respect to localization accuracy due to accumulated time drifts.

5.2. Big data processing

With the world-wide availability of low-cost and smart sensors wirelessly connected, sensors are being embedded in all sorts of devices of a variety of application domains including health, industry, environment, and household, enabling automation and making human life easier. Many organizations like IBM, The Central Nervous System for the Earth (CeNSE), and The Wireless World Research organization predict the growth of sensors to reach trillions by 2020. Such enormous number of sensors produce huge amount of data, which is growing exponentially and is expected to reach 40 zettabytes in 2020 [92]. In addition, as sensors are used in different areas and applications, they are sources of heterogeneous and diverse types of collected data, which can be either structured, such as temperature and pressure, or unstructured such as images, videos and audios. This results in the concept of big data that provides new opportunities, yet challenges, in what concerns processing and analysis to extract useful information. Indeed, WSNs provide useful big data that can be analyzed and processed to serve human life in several areas. For example, agriculture field [93] is complex and requires long-time experiences to make critical decisions such as predicting the yield of crops, which depends on environmental condition data and plant growth data. Researchers in [94] discussed the importance of big data in business field by allowing to make critical decisions, reduce risks, predict future outcomes and save money.

In the literature, several recent research works addressed the use of cloud computing technology to tackle big data challenges related to WSNs. In [95], the authors proposed a scalable and secure cloud-based architecture to collect, store and share huge amount of confidential data generated by medical body sensor networks allowing the healthcare institution to monitor patients’ health and improve rescue processes in emergency cases. Authors in [96] proposed a sensor-enabled patient monitoring system based on cloud computing concepts to enable remote monitoring of patients in order to reduce hospitals congestion. The proposed architecture is composed of three main servers including (1) patient information server or storage server which is responsible for storing terabytes of sensed data, (2) medical server and (3) communication server, which are designed to analyze the gathered data, detect and discover anomalies, and provide caretakers and medical specialists with real-time information about patients’ health status. Authenticated healthcare professionals and patients can consult this data via the Internet. The feasibility of the proposed architecture was validated with two scenarios including monitoring elderly people, and post-operative patients. In [97], the authors proposed a cloud-based architecture that automates the collection and processing of patient’s vital data and provides a real-time remote access to the medical staff. The wireless sensor nodes planted in patient’s bedside collect data and send it to the cloud to be stored and processed on virtual machines managed by OpenNebula cloud computing platform. The authors concluded from preliminary work that the architecture provides good performance. However, no performance evaluation were conducted to prove the efficiency of the architecture.

Researchers in [98] have designed and implemented an architecture that uses Web Services and cloud computing technologies to handle big data collected from WSNs. In [99], the authors surveyed existing tools and platforms for gathering and transmitting large volumes of sensor data to clouds such as MicroStrains, TempoDB, SensaTrack Monitor and Ostia Portus platforms. They identified a set of research challenges namely, communication, data exchange formats, security and interoperability. To address these challenges, the authors proposed a cloud-based sensor monitoring platform to collect, store, analyze and process big sensor data, and then send control information to actuators. The proposed architecture depicted in Fig. 3 uses the Generic Cloud Interface as an intermediate layer between sensor servers and the cloud to assure: (1) secure data access from and transfer to the cloud, (2) portability and interoperability, and (3) efficient data transport by formatting the sensor data into platform-neutral data exchange format, namely XML and JSON. In [100], researchers addressed the cloud-enabled large-scale WSNs aiming at integrating sensor networks with cloud infrastructures to manage Big Data. They used the Apache Hadoop framework to store and process big sensor data. In [101], the authors presented their vision for data quality centric big data infrastructure for federated sensor service clouds for what concerns collection, management and analysis. They have proposed the design of a cloud-based big data architecture to allow gathering, analyzing, storing and sharing large volume of data from large numbers of heterogeneous sensors. They also discussed a pricing model and Service Level Agreements (SLAs). Researchers harnessed big data technologies for storage and analysis of sensor massive feeds. They took benefit from: (1) NoSQL document store and key-value storage technologies to provide a scalable and efficient storage of historical sensor fed big data, and (2) Stream processing and Map-Reduce technologies for analyzing online and historical sensor data respectively. The architecture is Data Quality (DQ) relative to accuracy, availability and latency. But, no tangible implementation was implemented.

5.3. Sensor cloud virtualization

The recent research trends in WSNs aim at expanding the boundary of using them at large-scale, and integrating them in the daily human life by making objects surrounding us more proactive actors and able to interact and provide services. However, there are several handicaps that make WSNs not publically exposed and available to end-users at large-scale in particular; (1) deploying, managing and maintaining WSNs is usually complex and expensive, and requires a lot of detailed knowledge, (2) WSNs are traditionally designed to serve one application for a specific purpose.

With the advances in virtualization technologies, there is an emerging vision for using cloud computing to (1) promote the sharing of several physical sensors among multiple users seamlessly through common interfaces and, (2) provide abstraction layers on top of physical sensor devices to allow for their access. Sen-
Sensor clouds help in virtualizing physical sensors on a cloud computing infrastructure and exposing their sensing services via virtual sensors. Doing so, end-users would take benefit of a variety of remote sensing services without worrying about the locations of physical sensors or about their technical details such as management or maintenance as it is managed by the cloud.

SOA and Web Services played an important role in this process. In [102], the authors proposed an extensible cloud-based architecture for WSNs that uses Open.Sen.se platform, which is an open platform that provides APIs to (1) collect, analyze, store different sensor data streams using HTTP over IP; and (2) display and share collected sensor data using REST-based Web Services interfaces, thus enabling interoperable data access anytime from anywhere. The researchers validated their work by collecting temperature and battery voltage readings. They proved that collected data can be accessed anywhere from any mobile device within 11 seconds on average for the alert email notification and an efficient energy consumption on sensor nodes.

Researchers in [103] addressed the problem of physical sensor virtualization. They defined virtual sensor networks as a “sensor cloud that decouples the network owner and the user and allows multiple WSNs to interoperate at the same time for a single or multiple applications that are transparent to the user”. They have considered four different configurations of virtual sensors as presented in Fig. 4, and expressed as one-to-many, many-to-one, many-to-many, and derived configurations. It captures the cardinality of one physical sensor with respect to virtual sensors.

The authors also proposed a layered sensor-cloud architecture for the University of Missouri divided into three main layers: (1) the client-centric layer, which provides user interfaces and acts as a gateway between the user and the sensor cloud, (2) the middleware layer, which mediates between the users and physical sensors and performs the virtual sensors management and provisioning to end-users and clients, finally, (3) the sensor-centric layer, which directly interacts with the physical sensors.

The above work presents an interesting contribution to the field. However, they have used the old RMI technology for remote procedure calls, while the use of REST or SOAP Web Services would have been more appropriate for better interoperability and service exposure.

The Advanced Multi-risk Management SIGMA project [104] was proposed to manage heterogeneous data collected from various types of sensors and to provide services to prevent and manage risk situations. The authors used cloud computing to enable Sensor and Actuator as a Service (SAaaS), which consists of dynamically provisioning sensing resources (device-driven solution) and sensed data (data-driven solution) as services on demand according to users’ requirements. SAaaS architecture comprises three main modules: (1) the Hypervisor which provides abstraction and virtualization of sensing and actuation resources, (2) the Automatic Enforcer and (3) the Volunteer Cloud, which are responsible for the interaction among nodes. The authors have demonstrated the feasibility of their architecture by implementing a monitoring energy consumption application in an industrial environment and a smart traffic control.

In [105], researchers describe sensor cloud service environment which provides customers with remote services provided by a virtual sensor network. They addressed security aspects of the service.
delivery from three perspectives, including secure pre-deployment, secure pre-processing, and secure runtime. EPIKOUROS [106] is a virtualized platform aiming at integrating multiple heterogeneous sensor services in a cloud computing environment. The objective of the platform is to provide new methodologies and tools to design, build and deploy IoT applications on virtual environments and manage the business process procedures supported by the sensors’ infrastructure. The platform is based on the publish-subscribe model to collect sensor measurements. Following a SOA approach, the platform offers the required services to manipulate/manage collected information. It allows developers to choose its application deployment and the required sensors through a graphical user interface.

All the above works, among others, contributed to providing an insight on how to virtualize sensor networks and the benefits underlying such virtualization. However, all these efforts are still in its early phases and there remains plenty of room for more contributions and research to produce widely accepted solutions. In particular, there is a crucial need for standards to ensure interoperability at public and large-scale.

In Table 2, the sensor clouds architectures are compared with respect to the key requirements presented and discussed in Section 3.

<table>
<thead>
<tr>
<th>Work</th>
<th>Scalability</th>
<th>Reliability</th>
<th>Cost</th>
<th>Usability</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANGELS [87]</td>
<td>yes</td>
<td>–</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Integrating sensors with the cloud using dynamic proxies [88]</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>–</td>
<td>no</td>
</tr>
<tr>
<td>Mobile healthcare systems with multi-cloud offloading [90]</td>
<td>yes</td>
<td>–</td>
<td>–</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Energy efficient GPS sensing with cloud offloading [91]</td>
<td>yes</td>
<td>yes</td>
<td>–</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Healing on the cloud: secure cloud architecture for medical wireless sensor networks [95]</td>
<td>yes</td>
<td>–</td>
<td>–</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Web services framework for wireless sensor networks [98]</td>
<td>yes</td>
<td>–</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Towards a quality-centric big data architecture for federated sensor services [101]</td>
<td>yes</td>
<td>–</td>
<td>–</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Sensor-cloud: a hybrid framework for remote patient monitoring [96]</td>
<td>yes</td>
<td>yes</td>
<td>–</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>A cloud computing solution for patients data collection in health care institutions [97]</td>
<td>yes</td>
<td>–</td>
<td>–</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Integrating wireless sensor network into cloud services for real-time data collection [102]</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Sensor cloud: a cloud of virtual sensors [103]</td>
<td>yes</td>
<td>–</td>
<td>–</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>SensorCloud: an integrated system for advanced multi-risk management [104]</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>–</td>
<td>no</td>
</tr>
<tr>
<td>EPIKOUROS platform [106]</td>
<td>yes</td>
<td>–</td>
<td>–</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

6. Vehicular cloud

Despite many safety advancements such as airbags, cars remain more dangerous than other means of transportation. In the United States, between 2000 and 2009, accidents involving private users of cars and light trucks were the cause of 70% of the fatalities in transportation incidents [107]. Worldwide, road traffic accidents are the number one cause of death among those of age 15–29 [108]. Vehicular Ad-hoc Networks (VANETs) are being introduced to enable Cooperative Intelligent Transport Systems (C-ITS) that include not only information and comfort functions (such as weather and traffic conditions) but also safety critical applications, such as vehicle platooning, forward collision warning, and blind intersection warning systems. A VANET [109,110] is a self-organized network that can be formed by connecting vehicles equipped with Dedicated Short Range Communications (DSRC). As illustrated in Fig. 5, the communication can be between vehicles (V2V) and between vehicles and infrastructure (V2I), through Road-Side Units (RSUs).

In Europe, several car manufacturers signed a memorandum regarding the intention to make vehicles available with V2x technologies by 2015 [111], and, in the USA, the National Highway Traffic Safety Administration (NHSTA) expressed the intention to make connected vehicle technologies mandatory in light vehicles [112]. This large-scale deployment of VANETs brings important opportu-

nities to improve safety and comfort functions, and opens the door to new opportunities by merging cloud computing with VANETs. The Vehicular Cloud [113] is a new hybrid technology, which takes advantage of cloud computing to serve the drivers of VANETs by using resources, such as computing, storage and Internet for decision making.

The term Vehicular Cloud was firstly coined in [114] where Mohamed Eltoweissy et al. defined vehicular cloud as a group of largely autonomous vehicles whose corporate computing, sensing, communication and physical resources can be coordinated and dynamically allocated to authorized users.

In this section, we will review existing contributions to the development of the Vehicular Cloud (VC) concept by presenting: (1) VC Design Issues and VC Architecture, (2) Offloading Computation, (3) Big Data, and (4) Virtualization.

6.1. VC design issues and VC architectures

Both cloud computing and Vehicular Cloud have similarities in that they offer a pool of resources like storage, communication and computing. However, Vehicular Clouds have location-relevant applications, where the data is originated from the vehicles and its surrounding environment. Vehicular Cloud applications are also characterized by their mobility, as vehicles are constantly moving. Therefore, pooling of resources must cope with these dynamics.

Some research works have discussed Vehicular Cloud architectures and their requirements. For instance, authors in [113] have summarized a typical VC architectures into three layers: inside-vehicle, communication, and the cloud (see Fig. 6). The inside-vehicle part is responsible for monitoring the driver’s health and mood and collecting information such as temperature and pressure. The cloud can be used for storing information collected from sensors and performing complex computations. They have also discussed some application scenarios and how the cloud will serve
some VANETs applications in terms of resource allocation, remote access, storage capabilities, and security and privacy.

The work reported in [115] examines the evolution of VANETs with two emerging paradigms: Information-Centric Networking and Vehicular Cloud Computing. This work defines vehicular cloud computing as a set of resources, remotely accessed by vehicles to share their data and collaborate to produce some services. They propose to leverage Information Centric Networking (ICN) to propagate cloud contents between vehicles. This creates a Vehicular Cloud Network (VCN) to offer higher-level vehicular services: cloud resource discovery, cloud formation, task assignment and results collection, and content publishing and sharing.

The DARWIN architecture[116] builds on Service Oriented Architecture to design a next-generation automotive software platform that interacts with cloud services. The architecture is composed of two key service components: the Service Process Manager and the Service Space, interacting both inside and outside of the vehicle to form a vehicular cloud. From design perspective, the authors discussed three main challenges to be considered in the design of the architecture, namely, (1) real-time guarantees, (2) safety assurance security, and (3) software engineering issues. The DARWIN architecture has the benefit of being designed upon conventional automotive components (AUTOSAR) and standard messaging protocols, and follows a SOA approach to access and monitor services of automotive devices. It was demonstrated that REST web services reduce processing time up to 40% as compared to SOAP web services.

Another vehicle cloud computing architecture was proposed in [117], which includes a device level, a communication level and a service level. The architecture focuses on providing real-time vehicle cloud services such as road traffic monitoring, and health care monitoring. The main merit of this work is to address real-time services in vehicular networks. However, the weakness of this paper is that it is limited to the conceptual high-level design without a validation or concrete implementation of the architecture.

6.2. Offloading computation

Modern cars are well-equipped with a multitude of on-board computing devices that enable them to take advantage of cooperative functions that exploit VANETs, such as determining the traffic state, and the distance between neighboring vehicles. However, there is a great potential to improve and enable new functionalities by offloading computation intensive blocks from the vehicles to the cloud. Computations offloading in VANETs are of two kinds: (1) "between vehicles": a vehicle may send a part of an application to another more powerful car to be executed, and (2) between a vehicle and the infrastructure.

In the last few years, several research works discussed computation offloading in vehicles. Authors in [118] proposed and implemented a framework for offloading computation for interconnected vehicles. Vehicles in the proposed framework may select between four selection strategies, which were based on (1) the computation capability of each available surrogate vehicle, and (2) the longest time interval between inter-accessible vehicle pairs. The four selection strategies proposed in this paper are (1) Random selection strategy, (2) Computing capacity-based selection strategy, (3) Distance-based selection strategy, and (4) Multi-attribute selection strategy. Through simulation, the authors have shown that strategy (4) performs better. This work was targeting Vehicle-to-Vehicle offloading which can be useful in the case of absence of an RSU nearby.

Floating Car Data (FCD) is a method used to estimate the overall traffic conditions based on data (such as position of the vehicle, direction, and speed) generated by the cars and mobile phones of the users in the cars. Authors in [119] have presented the gain obtained by offloading FCD to a cloud environment in order to resolve the limitations when managing FCD data by a traditional cellular network paradigm. They have designed a Vehicle-to-Vehicle based FCD offloading model in which data is collected by a DSRC and transmitted to a set of selected vehicles. In turn, these vehicles gather the data they receive to upload it via a cellular network. This work supports the advantages of employing cloud offloading in such scenario. While the widespread of FCD-based services may increase the uplink load on the cellular networks, the authors have demonstrated that this approach may decrease traffic load and the cost of uploading data through local aggregation. Interested in FCD, Razvan Stanica et al. [120] have studied FCD offloading computation through vehicle-to-vehicle (V2V) communication. The main objective of this research was to identify a set of vehicles that gather and collect FCD data from their neighboring vehicles and
perform data fusion and upload it to the cloud. Through experiments, authors have proven that this approach achieved good performance.

The work presented in [121] is concerned with developing disaster management system for transportation systems called Intelligent Cloud-based Disaster Management System (ICDMS). The system exploits V2X communications to help vehicles exchange data collected from various locations into decisions and strategies. The proposed system has proven its effectiveness when compared with a disaster response system. However, these results are verified under very specific conditions (rate transmission, message size, etc.), which cannot prove the effectiveness of the system in real scenarios.

6.3. Big data processing

Today, with the revolution of communication technology, vehicles can communicate, collaborate, and connect with each other without human mediation. This cooperation may generate a huge volume of information when cars are sending their location, speed, and environment-related information. Catastrophic accidents have encouraged vehicle designers to analyze this increasing data to predict traffic and ensure drivers' safety. Big data for vehicles is about turning the ever-increasing collection of data from various agents into actionable information [122]. The challenge of this technology remains in collecting, transmitting, and analyzing this information and subsequently sending the results to all relevant target vehicles, all at high speed.

Lionel Nkenyereye et al. [123] presented a study of big data using hadoop framework to process diagnostic data of a set of connected vehicles. Diagnostic data is collected from android software, stored in datacenters, and processed by Hadoop. The outcomes are stored on web servers allowing third party access via web services. The system presented is a good example of how Big Data processing can be applied in the context of vehicular applications.

A novel multilayered vehicular data cloud platform was proposed recently [124]. The work introduces two vehicular data cloud services: an intelligent parking cloud service and a vehicular data mining cloud service. They present two different data mining techniques for the vehicular data mining cloud service, and experimentally show the feasibility of their vehicular data mining service. This is another interesting Big Data application, but unfortunately it did not present experiments to evaluate the quality of the services.

Yisheng Lv et al. describe in [125] a deep-learning method to predict traffic flow. Served from a rich amount of traffic flow related data, they have designed an architecture to extract traffic flow features. The architecture contains a logistic regression layer for traffic flow prediction. The accuracy and the effectiveness of the proposed method are well studied throw experiments and comparative study with previous works. Interested in vehicle navigation information big data analysis, authors in [126] have designed a model for providing the drivers the appropriate service. The proposed model is mainly composed of (1) a vehicle data collection module used to collect and transmit data to the collection center along with the user’s confirmation and, (2) the vehicle information center used to manage this data and offer driving information. One key advantage of the proposed model is the use of standard components like OBD-II. Unfortunately, the use of this standard is limited to a specific list of cars. Moreover, security concerns related to mobile devices should be considered. Authors in [127] have designed a Storage-as-a-Service system. Vehicles send data to a database created in roadside units. This data can be used later by other drivers or vehicle producers. The work present a parking mall scenario. But, no architectural view and components of the system were described. Shengcheng Yuan et al. in [128] were interested in analyzing large-scale traffic evacuations using big data techniques to provide better information support for emergency decision. They have designed a cross-simulation method that, when applied to a huge amount of collected data, can generate useful information for emergency decision. In a first step, this method collects the dynamic distance-speed relation for all the drivers. These data are then stored in a database. After calibration, the data distribution is analyzed to determine whether they would create an emergency situation. Finally, in the adaptation phase the system can estimate recent driving behavior states. Despite the idea of this research paper is important, it does not take into consideration the traffic and road conditions which may influence the acquisition phase.

We note that the systems presented were not designed to produce real-time answers that can be acted upon.

6.4. Virtualization

Virtualization has been used in networking for resource optimization, consolidation, maximizing uptime, workload migration, etc. But, recently the idea of virtual vehicle was introduced in the automotive world. This will enable the electrical system of the vehicle to integrate more functions with fewer control devices.

Virtualization already has its place in standard vehicular architectures such as AUTOSAR (Automotive Open System Architecture) [129], which enables the use of a component based software design model for the design of a vehicular system that can be realized through virtualization that isolates the different functional modules. AUTOSAR is a good solution to integrate components within an Engine Control Unit (ECU). However, the integration of software inside and outside the vehicles over different networks is a challenging issue.

Authors in [130] proposed a virtualization layer called the VaNetLayer, which defines procedures that enable mobile nodes to collaborate and create an infrastructure of virtual nodes to make the VANET a more reliable communication environment. The VaNetLayer introduced some modifications and additions to the Virtual Node Layer (VNLayer) proposed in [131] to cope with the inefficiency of the VNLayer constructs when tested with VANETs. Through simulations, the authors demonstrated the advantages of their work, in that it ensured good packet delivery ratios and better download time compared to torrent-like solutions.

Innovation in automotive industry has increased the number of electronic control units (ECU) used in cars. This has led to complexity problems in terms of energy consumption, cost, and installation space. To solve this problem, Simon Gansel et al. in [132] proposed to consolidate automotive systems through virtualization to share hardware between different components especially, graphics hardware. They have designed an architecture of a Virtualized Automotive Graphics System (VAGS) to isolate between graphics applications running in dedicated VMs. This work was based on a technical requirement that this architecture was derived from relevant ISO standards, which is a strong point in favor of its industrial adoption. However, the proposed architecture was implemented but no performance evaluation or comparative studies were presented.

Considering the importance of a vehicular gateway in linking in-vehicle networks with the Internet, Zonghua Gu et al. [133] proposed an automotive telematics gateway based on virtualization. The gateway supports two guest operating systems to separate between the executions of real-time and non real time applications. The architecture of the gateway performs real-time scheduling to meet the task management dynamicity. While several applications were developed on the gateway, no performance evaluation of the proposed gateway was presented.
Table 3
A summary of vehicular cloud architectures.

<table>
<thead>
<tr>
<th>Work reference</th>
<th>Scalability</th>
<th>Reliability</th>
<th>Cost</th>
<th>Usability</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARWIN architecture [116]</td>
<td>yes</td>
<td>yes</td>
<td>–</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Real time services for future cloud computing enabled vehicle networks [117]</td>
<td>yes</td>
<td>–</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ICMDs: an intelligent cloud based disaster management system for vehicular networks [121]</td>
<td>–</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>A study of big data solution using hadoop to process connected vehicle’s diagnostics data [123]</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Developing vehicular data cloud services in the IoT environment [124]</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>VaNetLayer: a virtualization layer supporting access to web contents from within vehicular networks [130]</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Towards virtualization concepts for novel automotive HMI systems [132]</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Design and implementation of an automotive telematics gateway based on virtualization [133]</td>
<td>–</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

In Table 3, the architectures for integrating vehicles with cloud computing are compared with respect to the key requirements presented and discussed in Section 3.

7. Discussions and future challenges

This paper presented an overview on the potential use of cloud computing to promote cyber-physical applications, and particularly focused on three major areas of applications, namely robotics, wireless sensor networks and vehicular networks.

Certainly, huge efforts have been carried out over the last five years to provide concrete examples on what a cyber-physical cloud can be. However, the path is still long for this technology to reach a certain maturity as there are several new challenges that must be addressed. Next, we summarize these challenges:

• **Need for standards**: For any system to be widely adopted, there is a crucial requirement to adopt standards to promote interoperability among heterogeneous systems and provide universal solutions that are vendor-independent and platform-agnostic. The current status of cyber-physical clouds clearly shows that most of the solutions are proprietary although some of them rely on open-source platforms, mainly Apache Hadoop with its Map-Reduce framework for data storage and processing. But this is not sufficient if universal agreements on networking protocols, data exchange formats, interfaces, etc, are not fulfilled. In other words, it is important to unify for each technology a set of standard protocols that govern the interaction between cyber-physical systems, Machine-to-Machine and also their interaction with clients and end-users, as well as the storage and processing processes, and network interfaces. The use of Web Services, whether REST or SOAP, seems to be very promising considering their widespread use and their efficiency in ensuring the integration of heterogeneous systems. Their use with cyber-physical systems can be easily extended to ensure the cyber-physical system to cloud integration.

• **Privacy and security**: There has been very little work on cyber-physical clouds security and privacy issues. Although several challenges would be shared with traditional cloud computing systems, cyber-physical clouds would be subject to additional threats that must be carefully addressed. In fact, considering the fact that cyber-physical clouds are highly distributed systems involving different and greatly heterogenous components, attacks can occur at different layers, either at the cyber-physical layer or the cloud layer. Therefore, end-to-end security mechanisms must be conceived and implemented to ensure the integrity of any transaction that takes place in the cyber-physical cloud. This is rather a crucial requirement as any vulnerability may have serious consequences on safety-critical applications. For example, an attacker may take over a physical sensor infrastructure and generate faulty data, then disrupt the system operation, or alternatively gain access to the cloud infrastructure and get access to unauthorized data. Security problems in clouds for sensor services were addressed in recent work [134] and also discussed earlier in [135]. On the other hand, privacy represents a major concern as data emanating from private cyber-physical system sources is stored in the cloud, which can be public or private, and presents the thread to be access by unauthorized tiers.

• **Real-time**: Cyber-physical clouds encompass time-sensitive applications that require real-time guarantees to deliver time-critical data, in particular for automation and control applications. For example, in mission-critical scenarios where sensor nodes populate the cloud with their data to take appropriate actions to be executed by actuators or robots, there is a need to have a bounded end-to-end delay to guarantee the correctness of the system operation. The concern becomes even more serious when data grows at large-scale. There is a need to carefully perform the dimensioning of the system in advance to ensure the quality of service needed for the operation of time-critical applications. Resource provisioning, scheduling and differentiated services are examples of possible research axes in this regard.

• **Programming abstractions**: Considering the big data scale in cyber-physical clouds, current programming abstractions might be insufficient to sufficiently scale with the size of the data to be processed. Although Map-Reduce framework represents a defacto standard for processing large data, there is still needs for companion mechanisms to filter out data coming from heterogeneous sources and thus reducing the computation space complexity through effective sampling without compromising the outcomes. In addition, it is important to provide developers with new Application Programming Interfaces (APIs) that allow them to easily interact with cyber-physical clouds, in the same way as public clouds, like Amazon Web Services, Google Apps Engine, Facebook and others, are providing for developers. There are currently some efforts and solutions in this respect like the Sensor API and Samsung GALAXY SDK for Android developers.

Acknowledgement

This work is supported by the Dronemap project entitled “DroneMap: A Cloud Robotics System for Unmanned Aerial Vehicles in Surveillance Applications” under the grant number 35–157 from King Abdul Aziz City for Science and Technology (KACST), and supported by Prince Sultan University.

References


The robot brain project, (http://robrobotbrain.net/).


Google goggles, (http://www.google.com/mobile/goggles/).


[129] Autosar (automotive open system architecture), (http://www.autosar.org/).
Rihab Chaâri is a PhD student in Department of Computer Science and Engineering at the National School of Computer Science and Engineering (ENSI). She is a member of The COINS (Cooperative Intelligent Networked Systems) research group. She received her master degree from the National Engineering School of Sfax in 2011. Her master thesis was on the performance evaluation of 6LoWPAN networks behavior and RPL protocol as well. Her current work focuses on designing cloud architecture for robotic applications. Research Interests: Cloud computing, cloud robotics, cooperative robots.

Fatma Ellouze is a PhD student in Department of Computer Engineering and Applied Mathematics (DGIMA) in National School of Engineers of Sfax (ENIS). She is a member of The COINS (Cooperative Intelligent Networked Systems) research group. She received her engineer diploma from the National School of Engineers of Sfax in 2013. Her thesis project focuses on virtualizing robotic services using cloud computing concepts.

Anis Koubaa received his B.Sc. in Telecommunications Engineering from Higher School of Telecommunications (Tunisia), and M.Sc. degrees in Computer Science from University Henri Poincare (France), in 2000 and 2001, respectively, and the Ph.D. degree in Computer Science from the National Polytechnic Institute of Lorraine (France), in 2004. He was a faculty member at Al-I mam University from 2006 to 2012. Currently, he is an associate professor in the Department of Computer Science at Prince Sultan University and research associate in CISTER Research Unit, ISEP-IPP, Portugal. He becomes a Senior Fellow of the Higher Education Academy (SFHEA) in 2015. He has published over 120 refereed journal and conference papers. His research interest covers mobile robots, robotics software engineering, Internet-of-Things, cloud computing and wireless sensor networks. Dr. Anis received the best research award from Al-I mam University in 2010, and the best paper award of the 19th Euromicro Conference in Real-Time Systems (ECRTS) in 2007. He is the head of the ACM Chapter in Prince Sultan University. His H-Index is 27.

Basit Qureshi is currently an assistant professor in the College of Computer & Information Science at Prince Sultan University. He received his PhD degree in Computer Science from University of Bradford, UK and Master of Science degree in Computer Science from Florida Atlantic University respectively. His research interests include Wireless Networks Security, Trust security and privacy (TSP) in wireless networks and TSP in P2P applications for Mobile Ad Hoc wireless Networks. To date he has published in various reputable international journals and conferences. He is a member of IEEE, IEEE Computer Society, IEEE Communication Society and ACM.

Nuno Pereira is a Professor at the School of Engineering of the Polytechnic of Porto (ISEP). The research by Nuno Pereira was published in more than 30 technical papers in peer reviewed scientific venues. He was involved in numerous international research projects and was the Principal Investigator of PATTERN (Programming AbsTractions for wireless sEnsOr Networks), and sub-Project Leader and the High-level Architect in DEWI (DEpendable Embedded Wireless Infrastructure). Recently, Nuno served as the Technical Program Co-Chair of EWSN 2015.

Dr. Habib Youssef received a Diplôme d’Ingénieur en Informatique from the Faculté des Sciences de Tunis, University of Tunis El Manar, Tunisia in June 1982 and a Ph.D. in computer science from the University of Minnesota Twin Cities, USA, in January 1990. He is a Professor of computer science at the ISTICom Hammam Sousse of the University of Sousse. Since October 2013, he has been serving as the Director General of Centre de Calcul El-Khawarizmi, Tunis, Tunisia, which is managing the Tunisian National Research and Education Network and providing Internet and application services to the Tunisian academic community. Dr. Youssef has over 200 publications in the form of books, book chapters, journal and conference papers. His current research interests are computer networks, performance evaluation of computer systems, and combinatorial optimization.
Eduardo Tovar holds since 1999 a PhD degree in electrical and computer engineering from the University of Porto, Porto, Portugal. Since 2003 he heads the CISTER Research Centre of the Polytechnic Institute of Porto (ISEP-IPP), an internationally renowned research centre focusing on RTD in real-time and embedded computing systems. He is deeply engaged in research on real-time distributed systems, multiprocessor systems, cyber-physical systems and industrial communication systems. He is currently the Vice-chair of ACM SIGBED (ACM Special Interest Group on Embedded Computing Systems) and was for 5 years, until December 2015, member of the Executive Committee of the IEEE Technical Committee on Real-Time Systems. Since 1991 he authored or co-authored more than 150 scientific and technical papers in the area of real-time and embedded computing systems, with emphasis on multiprocessor systems and distributed embedded systems. He has been consistently participating in top-rated scientific events as Program Chair or as General Chair. Notably he has been program chair/co-chair for ECRTS 2005, IEEE RTCSA 2010, IEEE RTAS 2013 or IEEE RTCSA 2016, all in the area of real-time computing systems. He has also been program chair/co-chair of other key scientific events in the area of architectures for computing systems and cyber-physical systems as is the case of ARCS 2014 or the ACM/IEEE ICCPS 2016.