 ISSN: 2321-8169
476 - 481

Open Cloud-Based PaaS Architecture for Service-Oriented Mobile Robots

Petri Oksa
Ph.D., Telecommunications Research Center
Tampere University of Technology
Pori, Finland
petri.oksa@tut.fi

Pekka Loula
Professor, Telecommunications Research Center
Tampere University of Technology
Pori, Finland
pekka.loula@tut.fi

Abstract—Open-source software frameworks such as Apache Hadoop and Robot Operating System (ROS) are helped researchers to reduce arduous engineering work releasing them to concentrate more on the core research. The impact of extensive knowledge repositories of robot operating system interconnected with a large distributed data storage framework entails enormous developmental step in the future robotic systems era. Hadoop and ROS integrate the key frameworks presented in this paper to obtain the open source architecture. The main subject of research, an approach to apply service-oriented robotic (SOR) cloud enabling PaaS (Platform-as-a-Service) mobile multi-robot architecture, is presented. Moreover, the focus is based on cloud-communicating service robots with lightweight workload requirements for the purpose of offloading heavy computation into a cloud. In an integrated experiment, the Ubuntu Linux environment is experimented after installing Hadoop cloud computing platform and ROS.

Keywords*-robot operating system (ROS); Apache Hadoop; open source; service-oriented robot; cloud robotics; mobile robotics*

*****

I. INTRODUCTION

Goldberg et al. announced industrial robot which was connected to the Web in 1994, allowing users to teleoperate the robot via an Internet browser [1], [2], [3]. Later on, several other Web-based teleoperated robots were developed and the IEEE Robotics and Automation Society established a Technical committee on Networked robots in May 2001 [1], [2], [3], [4]. James Kuffner, Research Scientist at Google, envisions a future when robots will feed data into a "knowledge database", where they will share their interactions with the world and learn about new objects, places, and behaviors [5]. In 2010, he introduced the term "Cloud Robotics" to describe an approach where Internet provides massively parallel computation and sharing of vast data resources [3], [6].

One of the cloud services is Apache Hadoop presenting open-source software for reliable, scalable and distributed cloud computing [7]. Regarding to cloud computing, the network and communication protocols are limited to a few standards such WSDL, SOAP, HTTP, and RESTful architecture [7], [8].

Apache Hadoop is a software framework for running applications on large clusters built of commodity hardware. Hadoop provides a distributed file-system (HDFS) and a parallel processing framework based on the MapReduce programming paradigm. The advantage of Hadoop is that it allows programmers to deal with Big Data while hiding the operational complexity of the parallel execution across hundreds of servers [9].

HDFS has a master/slave architecture. HDFS cluster consists of a single NameNode that manages the file system namespace and DataNode that store the actual data [10], [11]. Figure 1 shows the architecture of HDFS [10].

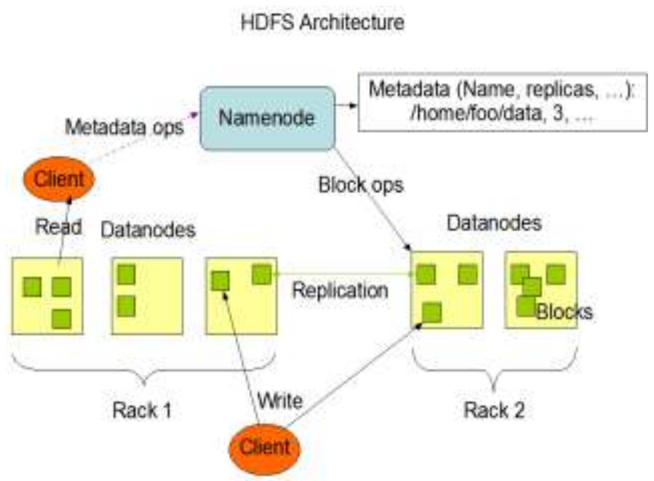


Figure 1. Architecture of the HDFS.

Today, mobile cloud robotics initiatives have recently emerged such as A'STAR Data Storage institute of Singapore proposing a software framework (ASORO) (http://www.a-star.edu.sg/asoro/), Google (Google Object Recognition Engine) [12], [13] DAvinCi and Researchers at the Laboratory of Analysis and Architecture of Systems, in Toulouse (http://www.laas.fr/2-30649-About-LAAS.php). One of the pioneering initiatives is European project RoboEarth proposing the implementation of Rapyuta [14], World Wide Web-like open source network for robots using a PaaS (Platform-as-a-Service) [15] service model. Rapyuta is based on an elastic computing model that dynamically allocates secured computing environments for robots [16]. The DAvinCi (Distributed Agents with Collective Intelligence) [17] is a cloud computing architecture for service robots offloading data intensive and computationally intensive workloads from the onboard resources on the robots to a backend cluster system. The

DAvinCi framework (see Fig. 2) combines the distributed ROS architecture, the open source Hadoop Distributed File System (HDFS) and the Hadoop Map/Reduce Framework. Also the FastSLAM algorithm in Map/Reduce is implemented to evaluate execution times to build a map of a large area with eight-node Hadoop cluster [17]. Kehoe, Matsukawa, Candido, Kuffner and Goldberg presented a system architecture, prototype, and experimental data on a cloud robotics system for recognizing and grasping common household objects [12]. Object recognition is performed off-board the robot using a variant of the Google Object Recognition Engine [13].

In order to achieve real-time and time-sensitive applications, the adoption of cloud computing is essential [18]. To provide fast and flexible access to data repository, the most of the data-processing functions is processed in the cloud [18].

The remainder of this paper is structured as follows: In Section II cloud-based service architectures are outlined. Section III describes ad-hoc cloud formed by a group of networked robots and an infrastructure cloud. Cloud infrastructure set-up and methods are described in Section IV. Finally, the paper is concluded in Section V with directions of the future work and research.

II. CLOUD-BASED SERVICE ARCHITECTURES

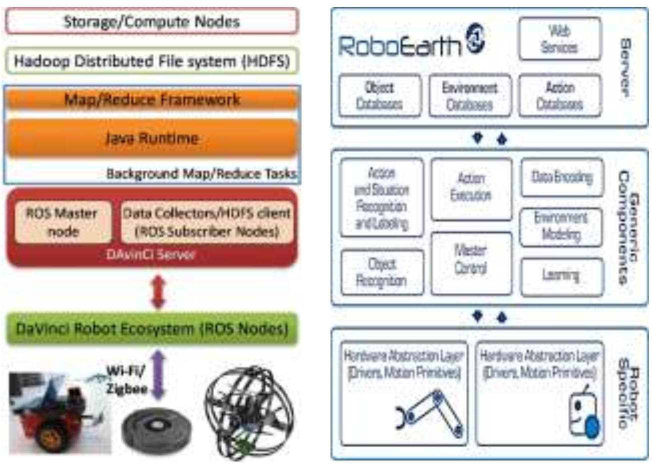
In the past few years, a few major cloud PaaS platform services are implemented, including Amazon Elastic Compute Cloud (EC2) [19], OpenShift (OpenShift Origin is an open source PaaS-based software) [20], Cloud Foundry (open source PaaS) [21], Google's App Engine (GAE) [22], and Microsoft Azure [23]. These cloud services, especially the popular GAE, are not so well-suited for robotics applications due to the limited subset of program API's designed specifically for Web applications, allowing only a single process, and does not expose sockets, which are indispensable for robotic software frameworks such as ROS [14], [16]. Other obstacle involves to a preconfigured set of processes running inside a computing environment that has only an HTTP connection to the outside [14]. This can be mainly explained about differences between Web and robotics applications, including programming languages, processes, and the communication protocols [16].

There are five cloud service delivery models: *Software-as-a-Service (SaaS)*, *Platform-as-a-Service (PaaS)*, *Infrastructure-as-a-Service (IaaS)*, *Telecom-SaaS (TSaaS)* [24] and *X as a Service (XaaS)*, where X can be any items, all deployed as public, private, community, and hybrid clouds [25], [26], [27]. The examples of XaaS include:

- Hardware as a Service - HaaS,
- Communication as a Service - CaaS,
- Databases as a Service - DBaaS,
- Identity Management as a Service - IMaaS,
- Desktop as a Service - DaaS, and
- Robot as a Service - RaaS

In SaaS, users are provided access to application software and databases. Cloud providers manage the infrastructure and platforms that run the applications. PaaS cloud computing platform typically include operating system, programming language execution environment, database, and Web server. IaaS offers computer – physical or virtual machines (VM) – and other resources. Users possibly have a limited control of select networking components, e.g. host firewalls. [15] TSaaS presents SaaS variant for telecom public cloud, which introduces TSaaS-engine to drive the multi-tenancy for carrier grade telecom applications outside the actual business logic [24]. One of the common definitions of HaaS proposes a physical hardware on-demand. With a private HaaS cloud, hardware can be accessed as a resource within one company to make it available to all employees [28]. CaaS is an outsourced enterprise solution that can be leased from a single vendor [27]. CaaS can include VoIP (or Internet telephony), instant messaging (IM), collaboration, and videoconference applications using fixed and mobile devices [29], [30]. DaaS refers to outsourcing the database by back up, restore, availability, and storage management [31]. Some cloud platform providers offer DBaaS without physically launching a virtual machine instance for the database [32], [33]. IDaaS offer a cloud-based identity management solution that solves password sprawl and secures users accessing cloud and mobile apps [34]. In DaaS, a cloud service provide desktop environment of virtual machine running on the cloud [35]. The DaaS provider typically takes full responsibility for hosting and maintaining the compute, storage, and access infrastructure, as well as applications and application software licenses needed to provide the desktop service in return for a fixed monthly fee [35], [36], and [37]. The basic elements of RaaS fulfill functions as service provider, a service broker, and a service client [38], [39]. Furthermore, RaaS should meet the requirements to complete functions of SOA, the common term in cloud services, which is explained later in this chapter.

DAvinCi and RoboEarth both are PaaS-based systems built with Hadoop and ROS that shares a set of algorithms and data within a robotic network (see Fig. 2 for systems comparison) [40]. Each unit can share its sensor data and also upload data to be processed in the cloud [40]. For instance, the robots are able to perform simultaneous localization (SLAM) and generation of their 3-D environment maps using the cloud infrastructure instead of using their onboard systems [17], [40]. SLAM exposes cloud service that can be accessed e.g. by the Internet through ROS messages encapsulated in HTTP requests/responses [6]. In [41], the concept of elastic cloud computing architecture for cloud robotics is proposed and its characteristics are discussed in Section III.



(a) (b)

Figure 2. Comparison of (a) DavinCI and (b) Rapyuta architectures.

 ISSN: 2321-8169
476 - 481

Figure 3 outlines the structure of *PaaS* delivery model. PaaS gives the capability to deploy consumer-created or acquired applications using programming languages and tools supported by the provider [25], [42].

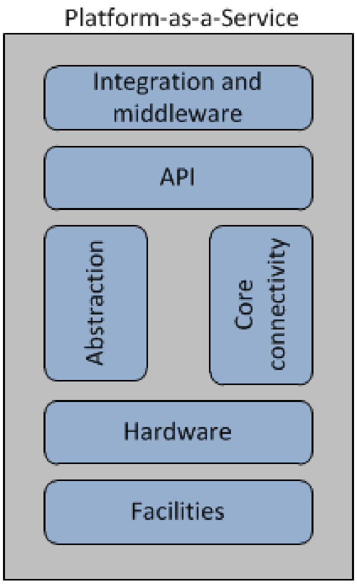


Figure 3. The structure of PaaS delivery model.

There are two main robotic developers, ROS (Robotic Operating System) toolset and Microsoft toolset known as a Robotics Developer Studio (Microsoft RDS, MRDS) utilizing Web services [43]. RDS is a Windows- and .NET-based environment for robot control and simulation. MS Toolset allow to develop an MRDS application contain a graphical environment (MS Visual Programming Language, VPL) in C#, and 3D simulation tools. RDS is not promoting the use of traditional Web service standards, such as the Web Service Description Language (WSDL), whereas ROS conceptually provide the ability to create a peer-to-peer network of processes that process data together [43].

In cloud robotics, cooperation of various web services is possible, and the technology that underlies the global image of system structure is called Service-Oriented Architecture (SOA) [44]. Remote Procedure Call (RPC) is one example of Service-Oriented Architecture (SOA). RPC is a communication protocol that allows a process to execute a procedure in another process [14]. Related to practical experience of SOA, whose researchers are active in the field of Cloud Computing technology, it has been found ideal for the control and management of industrial robots [38], [39], [44]. Moreover, it is researched that SOA can offer highly efficient learning, which is not possible with stand-alone robots [44], [45], [46]. This is a significant benefit considering the development work of cloud-based robotics.

Service-Oriented Robots (SOR) is an architectural paradigm posing the end-user in the SOA (Service-Oriented-Architecture) [47], [48], [49]. From the service model taxonomy, SOA is categorized to SaaS delivery model (see section 2.1) [50], [51].

In [6] Cloud Robotics is expected to develop Robotics in five different topics: *Big Data*, *Cloud Computing*, *Open-Source/Open-Access*, *Collective Robot Learning* and *Crowdsourcing and call centers*. Below, the key features of these five topics are described:

Big Data

Robots can generate vast amounts of images, maps, and object data by using high-throughput sensors including 2D image features, 3D features, and 3D CAD models in an online database [6], [12], [52].

Cloud Computing

Computing power is one of the most important requirements of cloud computing services, especially parallel grid computing on demand for statistical analysis, learning, and motion planning [6], [12], [52].

Open-Source/Open-Access

Code, data, algorithms, benchmarks, and hardware designs can be facilitated by the Cloud. One primary example is ROS, which provides libraries and tools to help software developers create robot applications. [6], [12], [52].

Collective Robot Learning

The Cloud allows robots and automation systems to share trajectories, control policies, and outcomes. For example, all the environmental data that robots have collected, is then available for collective learning tasks by the Cloud. [6], [12], [52].

Crowdsourcing and call centers

Errors and exceptions are detected by robots and automation systems, which then access human guidance offline and on-demand at remote call centers [6], [12]. Cloud based services allow developers to share their problems and contribute to each other developments, or to develop robotic system in a crowdsourcing paradigm where all contribute with their knowledge, and work to the final robotic system [52].

III. CLOUD COMPUTING ARCHITECTURES

Hu, Tay and Wen [41] proposed ad-hoc cloud formed by a group of networked robots and an infrastructure cloud. Three of elastic computing models are described (Fig. 4):

- *Peer-Based Model*
- *Proxy-Based Model*
- *Clone-Based Model*

In *Peer-Based Model* each robot or virtual machine (VM) forms a computing unit. The *Proxy-Based Model* connects group of networked robots in which one robot operates as a group leader. Group leader robot communicates with a proxy VM in the cloud infrastructure, bridging the interaction between the robotic network and the cloud. All of the computing units are organized into a two-tier hierarchy. In the *Clone-Based Model* each robot has a corresponding system-level clone in the cloud. Either robot or its clone can execute a task.

The set of robotic clones form a peer-to-peer network causing better connectivity than the physical ad-hoc M2M network. This model allows for sporadic outage in the physical M2M network [41], [53]. As illustrated in Fig. 5, the cloud robotics architecture is organized into two tiers: a machine-to-

ISSN: 2321-8169
476 - 481

machine (M2M) level and a machine-to-cloud (M2C) level [41]. M2M level means that a group of robots form a collaborative group and M2C level the infrastructure cloud provides shared computation and storage resources that can allocated for real time demand. Thus, M2C level robots can offload their computation-intensive tasks for remote execution [41], [53].

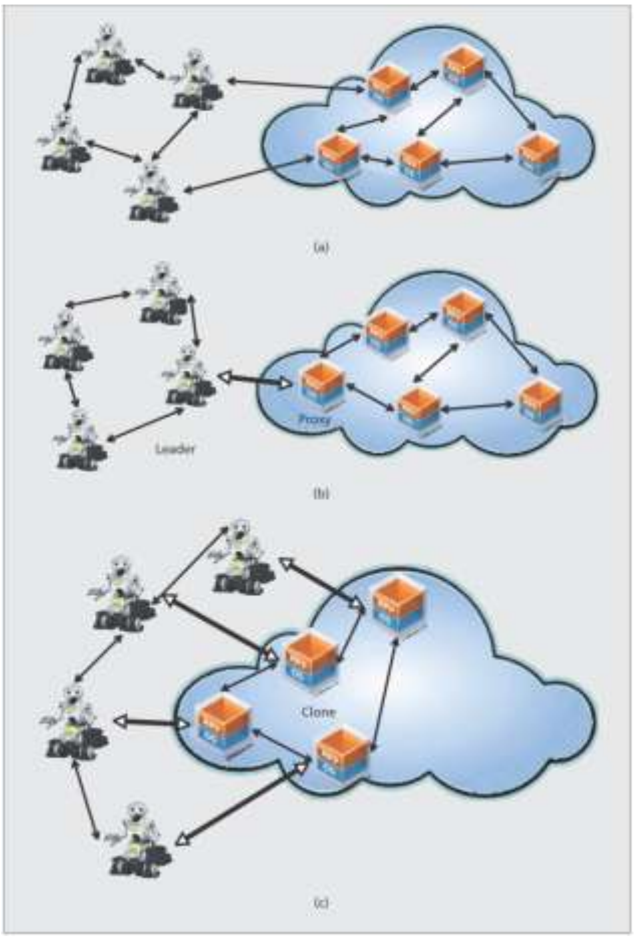


Figure 4. Three network models for cloud robotics: (a) peer-based; (b) proxy-based; (c) clone-based model [41].

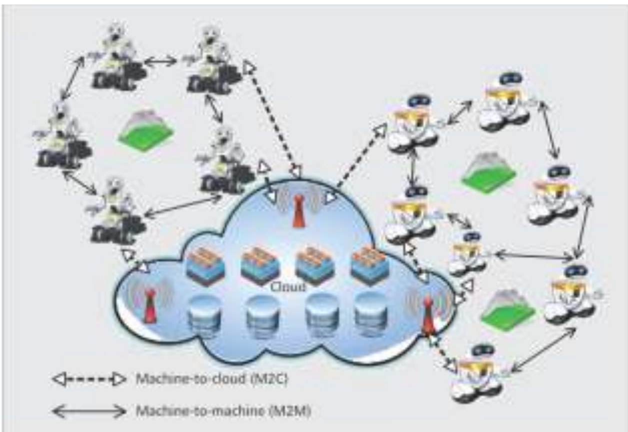


Figure 5. M2M/M2C communications, sharing their resources in ad-hoc cloud and communicating with infrastructure cloud [41].

IV. Cloud Infrastructure Set-Up

In this section installing of generic Hadoop cloud infrastructure and ROS is described and experimented from a starting point. The purpose of the set-up is to tackle out errors and exceptions and to help practitioners familiarize with open source cloud data storage facilities.

A. *Single Node Cluster Set-Up for Hadoop*

At first, Ubuntu Linux 14.04 LTS operating system is installed to a common PC, either desktop or laptop. After downloading and installing Hadoop, the latest Java version (located in `.bashrc` file) should be updated and carefully checked that the script file is saved into a desired directory path, otherwise `ssh localhost` and Hadoop `NameNode` will not start properly. Figure 6 shows Hadoop NameNode and MapReduce single node set-up after configurations.

Figure 6. Hadoop NameNode and MapReduce.

After installing Hadoop all the Ubuntu Linux services must be checked for the avoidance of loading up the `error_log`. For instance Common UNIX Printing System (Cups) can harmfully use the whole capacity of hard disk drive via `error_log`. This can be avoided by stopping the Cups daemon and by following the filesystem log files:

```
robo@robo:~$ su root
Password:
root@robo:/home/robo# df -h
Filesystem   Size   Used   Avail Use% Mounted on
/dev/sda1    290G   9,8G   266G  4%/
```

```
total 222248640
drwxrwxrwx  2 root root    4096          syys   9 11:10 .
drwxrwxrwx 14 root syslog  4096          syys  15 12:00 ..
-rw-r-----  1 root adm     336           elo   27 12:36 access_log
-rw-rw-rw-  1 root adm     143           elo   22 09:14 access_log.1.gz
-rw-rw-rw-  1 root adm     126           elo   20 11:57 access_log.2.gz
-rw-rw-rw-  1 root adm     127           elo   14 12:15 access_log.3.gz
-rw-r-----  1 root adm     227582541324  syys  15 12:45 error_log
-rw-rw-rw-  1 root adm     138           elo   27 09:39 error_log.1.gz
-rw-rw-rw-  1 root adm     122           elo   26 09:38 error_log.2.gz
-rw-r-----  1 root adm     0             elo   14 12:14 page_log
```

B. ROS Installation

The latest version of ROS is Indigo Igloo (release date July 22nd 2014). When Ubuntu Linux and Hadoop are at the ready, working with ROS is quite straightforward. Figure 7 shows the ROS master filesystem (Debian packages) and basic repositories for Ubuntu through the downloaded Indigo distribution.

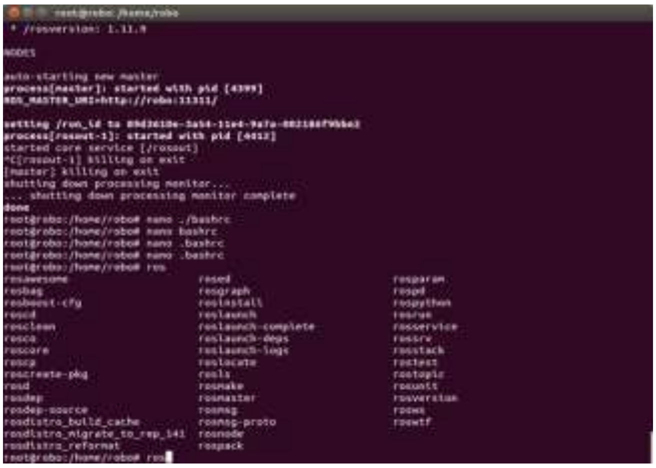


Figure 7. ROS master filesystem.

For beginner level users, ROS tutorial give the possibility get used to installing and configuring ROS environment on a PC. ROS has repositories e.g. for robot simulation, visualization and Android mobile operating system overall repository number being nearly 6000 to date. There are over 45 mobile robots that use or can be used with ROS software.

V. CONCLUSIONS

This paper presented service-oriented robotic (SOR) cloud enabling PaaS mobile multi-robot architecture. In cloud infrastructure set-up, the generic Hadoop cloud and ROS are installed and evaluated on Ubuntu Linux environment. The set-up experiment shows how to tackle out errors and exceptions in a fresh installation of Hadoop and ROS.

To take the next step in our future research plan, we have developed OCRP architecture platform (Open CloudRobotic Platform for Mobile- and Service-Oriented Ecosystem) providing commodity data management resources utilizing the connectivity with Apache Hive data warehouse and Apache Spark data processing facilities.

Currently we are designing and implementing Hadoop and ROS cloud robotics platform by using TurtleBot 2 Montado open source robot development kit and GROMA mapping robot. Both robots are equipped with Raspberry Pi Model B+ on-board computers. With two interconnected mobile robots, we experiment M2M and M2C level communications sharing their resources in cloud and communicating with infrastructure cloud.

REFERENCES

[1] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley, "Desktop teleoperation via the world wide web", Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 1, pp. 654-659, May 1995.

[2] Goldberg, K. (1994), "Beyond the web: Excavating the real world via mosaic", In Second International WWW Conference.

[3] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A Survey of Research on Cloud Robotics and Automation", in IEEE Transactions on Automation Science and Engineering, March 2014.

[4] K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler and G. Smith, "Collaborative teleoperation via the internet", in IEEE International Conference on Robotics and Automation,. Proceedings. ICRA'00, Vol. 2, pp. 2019-2024, 2000.

[5] J. Kuffner, "Robots with their heads in the clouds", Discovery News, 2011.

[6] K. Goldberg and B. Kehoe, "Cloud robotics and automation: A survey of related work," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-5, January 2013.

[7] Y. Chen, Z. Du & M. García-Acosta, "Robot as a service in cloud computing", Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), pp. 151-158, June 2010.

[8] Y. Chen and H. Hu, "Internet of intelligent things and robot as a service", Simulation Modelling Practice and Theory, 34, 159-171, 2013.

[9] D. Regvart, Y. D. UvA, S. Filiposka, M. de Vos, and T. Karaliotas, "Milestone MS101 (MJ1. 2.1): Network Architectures for Cloud Services", White Paper.

[10] The Apache Software Foundation, "HDFS Architecture", URL: <hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>, last accessed August 2014.

[11] ICT Call 4 RoboEarth Project 2010-248942, "Deliverable D6.1: Complete specification of the RoboEarth platform", December 2010, URL: <www.roboearth.org/wp-content/uploads/2011/03/D61.pdf>, last accessed August 2014.

[12] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine", in 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 4263-4270, May 2013.

[13] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner and K. Goldberg, "Grasping with Google Goggles". in Robotics and Automation (ICRA), 2013.

[14] G. Mohanarajah, D. Hunziker, R. D'Andrea and M. Waibel, "Rapyuta: A Cloud Robotics Platform", 2014, in press.

[15] P. Mell and T. Grance, "The NIST definition of cloud computing", 2011

[16] D. Hunziker, M. Gajamohan, M. Waibel and R. D'Andrea, "Rapyuta: The roboearth cloud engine", in 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 438-444, May 2013.

[17] R. Arumugam, V. R. Enti, L. Bingbing, W. Xiaojun, K. Baskaran, F. F. Kong and G. W. Kit, "DAvinCi: A cloud computing framework for

service robots”, in 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 3084-3089, May 2010.

[18] Cisco, “Cisco Global Cloud Index: Forecast and Methodology, 2012–2017, April 2014, URL: <www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf>, last accessed August 2014.

[19] Amazon Web Services, URL: <aws.amazon.com/ec2/>, last accessed August 2014.

[20] OpenShift, URL: <www.openshift.com/>, last accessed August 2014.

[21] Cloud Foundry, URL: <cloudfoundry.org/index.html>, last accessed August 2014.

[22] Google’s App Engine, URL: <developers.google.com/appengine/>, last accessed August 2014.

[23] Microsoft Azure, URL: <azure.microsoft.com/en-us/>, last accessed August 2014.

[24] S. Pal and T. Pal, ”TSaaS—Customized telecom app hosting on cloud”, in IEEE 2011 5th International Conference on Internet Multimedia Systems Architecture and Application (IMSAA), pp. 1-6, December 2011.

[25] D. C. Marinescu, “Cloud Computing: Theory and Practice. Newnes”, 2013.

[26] Z. Du, W. Yang, Y. Chen, X. Sun, X. Wang, and C. Xu, “Design of a robot cloud center”, in IEEE 2011 10th International Symposium on Autonomous Decentralized Systems (ISADS), pp. 269-275, March 2011.

[27] R. C. Garcia and J. M. Chung, “XaaS for XaaS: An evolving abstraction of web services for the entrepreneur, developer, and consumer”, in 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), (pp. 853-855), August 2012.

[28] A. Stanik, M. Hovestadt and O. Kao, “Hardware as a Service (HaaS): The completion of the cloud stack”, in 2012 IEEE 8th International Conference on Computing Technology and Information Management (ICCM), Vol. 2, pp. 830-835, April 2012.

[29] Interactive Intelligence, URL: <caas.tmcnet.com/>, last accessed August 2014.

[30] Microsoft, URL: < msdn.microsoft.com/en-us/library/bb896003.aspx>, last accessed August 2014.

[31] O. Ünay and T. İ. Gündem, “A survey on querying encrypted XML documents for databases as a service”, ACM SIGMOD Record, 37(1), 12-20, 2008.

[32] Wikipedia, URL: <en.wikipedia.org/wiki/Cloud_database>, last accessed August 2014.

[33] ScaleDB, URL: <www.scaledb.com/dbaas-database-as-a-service.php>, last accessed August 2014.

[34] Centrify, URL: <www.centrify.com/solutions/identity-as-a-service-idaas.asp#>, last accessed August 2014.

[35] S. Kibe, T. Koyama and M. Uehara, “The evaluations of desktop as a service in an educational cloud”, in 2012 IEEE 15th International Conference on Network-Based Information Systems (NBiS), pp. 621-626, September 2012.

[36] VMware, URL: <www.vmware.com/products/daas/features>, last accessed August 2014.

[37] Wikipedia, URL: <en.wikipedia.org/wiki/Desktop_virtualization#Desktop_as_a_Service>, last accessed August 2014.

[38] Y. Chen and H. Hu, “Internet of intelligent things and robot as a service”, Simulation Modelling Practice and Theory, 34, pp. 159-171, 2013.

[39] Y. Chen, Z. Du and M. García-Acosta, “Robot as a service in cloud computing”, in 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), pp. 151-158, June 2010.

[40] A. Chibani, Y. Amirat, S. Mohammed, E. Matson, N. Hagita and M. Barreto, “Ubiquitous robotics: Recent challenges and future trends”, Robotics and Autonomous Systems, 61(11), 1162-1172, 2013.

[41] G. Hu, W. P. Tay and Y. Wen, “Cloud robotics: architecture, challenges and applications”, IEEE Network, 26(3), 21-28, 2012.

[42] C. Alliance, “Security guidance for critical areas of focus in cloud computing v3.0”, Cloud Security Alliance, 2011.

[43] S. L. Remy and M. B. Blake, “Distributed service-oriented robotics”, IEEE Internet Computing, 15(2), pp. 70-74, 2011.

[44] R. Tsuchiya, S. Shimazaki, T. Sakai, S. Terada, K. Igarashi, D. Hanawa, and K. Oguchi, “Simulation environment based on smartphones for Cloud computing robots”, in IEEE 2012 35th International Conference on Telecommunications and Signal Processing (TSP), pp. 96-100, July 2012.

[45] J. Quintas, P. Menezes and J. Dias, “Cloud robotics: towards context aware robotic networks”, in International Conference on Robotics, pp. 420-427, November 2011.

[46] J. Kuffner, “Robots with their heads in the clouds”. Discovery News, 2011.

[47] W. Adiprawita and A. R. Ibrahim, “Service oriented architecture in robotic as a platform for cloud robotic (Case study: human gesture based teleoperation for upper part of humanoid robot)”, in IEEE 2012 International Conference on Cloud Computing and Social Networking (ICCCSN), pp. 1-4, April 2012.

[48] R. Doriya, P. Chakraborty and G. C. Nandi, “Robotic Services in Cloud Computing Paradigm”, in IEEE 2012 International Symposium on Cloud and Services Computing (ISCOS), pp. 80-83, December 2012.

[49] R. Doriya, P. Chakraborty and G. C. Nandi, ‘Robot-Cloud’: A framework to assist heterogeneous low cost robots”, in IEEE 2012 International Conference on Communication, Information & Computing Technology (ICCICT), pp. 1-5, October 2012.

[50] J. Namjoshi and A. Gupte, “Service oriented architecture for cloud based travel reservation software as a service”, in IEEE International Conference on Cloud Computing, 2009. CLOUD'09, pp. 147-150, September 2009.

[51] W. T. Tsai, X. Sun and J. Balasooriya, “Service-oriented cloud computing architecture”, in IEEE 2010 Seventh International Conference on Information Technology: New Generations (ITNG), pp. 684-689, April 2010.

[52] R. L. Aguiar, D. Gomes, J. P. Barraca and N. Lau, “CloudThinking as an Intelligent Infrastructure for Mobile Robotics”, Wireless Personal Communications, 76(2), 231-244, 2014.

[53] D. Lorencik and P. Sincak, “Cloud Robotics: Current trends and possible use as a service”, in 2013 IEEE 11th International Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 85-88, January 2013.