Cloud Robotics using ROS

Adarsha Kharel
SMIT
Majitar
East Sikkim

Dorjee Bhutia
SMIT
Majitar
East Sikkim

Sunita Rai
SMIT
Majitar
East Sikkim

Dhruba Ningombam
SMIT
Majitar
East Sikkim

ABSTRACT
The application of the cloud computing concept to robots is called Cloud Robotics. It is a concept that utilizes the services of the cloud so that robots can have learning abilities. Since applications for Cloud Robotics have to be developed in a platform, majority of the cloud application developers choose ROS for it. Robot Operating System (ROS) is an open source middleware that has a collection of inter-programming language headers to allow the sharing of data between independent programs. ROS provides a graph-like structure for cloud robotics. A new library for ROS that is a pure Java implementation, called rosjava, allows Android applications to be developed for robots. Since Android has a booming market and billion users, it would be a huge leap in the field of Cloud Robotics.

General Terms
Cloud Robotics

Keywords
Cloud Computing (CC), Cloud Robotics (CR), Personal Robot (PR) 2, Robot Operating System (ROS)

1. INTRODUCTION
Networked Robotics has a long history but Cloud Robotics is a relatively recent term. "Robotics on Cloud" is a concept that utilizes data collected from robots and sensor networks by creating cloud storage to save the information and reusing it at the time of need. It uses the help of the Internet to increase a robot's capabilities by reducing on-board computation and providing services on demand. Since robots will be connected to the cloud, they can communicate with each other and exchange useful information among each other. This off-board computing is called Cloud Computing. Massive parallel computation and real-time sharing of data resources are the basic services provided by Cloud Computing.

Since our discussion is on robots and the development of Robotic applications, we must know what a Robotic middleware is. There is an abstraction layer that resides between the operating system and the software application which is designed to manage the heterogeneity of the hardware, improve software application quality, simplify software design, and reduce development costs. This is known as a robotic middleware. ROS is one such middleware. Elkady and Sobh [3], in their discussion on Robotics Middleware, described the objectives of ROS as a middleware which provides services of the operating system such as low-level device control, hardware abstraction, message-passing between processes, and many others. So, ROS is not an operating system but a "metaoperating system" [13] for robot software which consists of a number of small tools to allow sharing of data between independent programs. ROS is an open source robot operating system that provides:

- Hardware abstraction.
- Low-level device control.
- Message Passing.
- Package Management.

2. CLOUD ROBOTICS
Generally, all the information that a robot has is contained within itself and there is very little scope for learning new things. So, there is a need of storing robot information in the Internet so that new things can be learnt easily and application for robots can share common features. This leads us to the concept of "Cloud Robotics."

The term "Cloud-enabled robotics" was presented by James Kuffner for the first time at the IEEE RAS International Conference on Humanoid Robotics in 2010. Now, we know it more commonly as Cloud Robotics (CR) [13]. There are 4 basic sequence of robotic actions associated with Cloud Robotics [as performed by Willow Garage's PR2 robot].

- Data capturing by the robot: In this step, there is a 2D camera and a 3D scanner which detects the object on-board of the robot. The output is a 2D image and a 3D point cloud, but only the 2D image is sent for processing to the cloud server.
- Data Processing on cloud servers: The server returns a 3D CAD model of the object to the robot if it was able to identify the object based on the 2D image.
- Processed data is applied: In this step, the robot determines the object's position and orientation based on the obtained 3D CAD model and the measured 3D point cloud. The robot, then, applies its functionality thereafter.
- Sending feedback to the server: Finally, the robot sends the results of the functionality back to the server so that the server's database can be trained for further use.

Cloud Robotics has the prospective to considerably advance a robot's working in human environments. Currently, under the mainstream research of robotics and artificial intelligence (AI), Cloud Robotics is considered to be one of

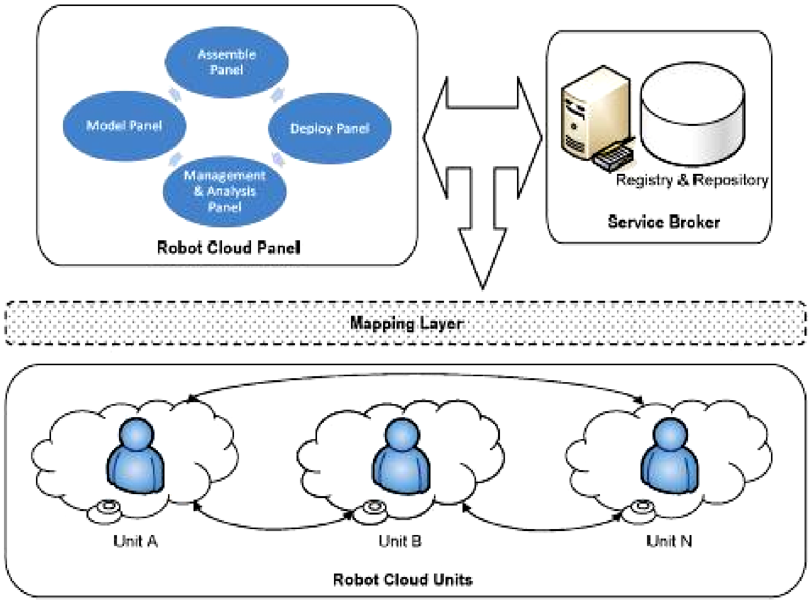

Fig: Robot Cloud Center Architecture, taken from reference # [21]

the most looked-upon research fields [11]. The existing cloud-computing infrastructure is based upon a service-oriented architecture. The key focus of the research on Cloud Robotics is to look at the likelihood of using this existing infrastructure. For various robotic applications of the future, the main aim in this field is to establish a shared network resource in order to make robots more scalable.

3. CLOUD COMPUTING

Commonly, Computing Clouds [18] are classified into 3 types: Public Clouds, Private Clouds, and Hybrid Clouds. Another type of cloud is Community Cloud. When a Cloud is offered to the common public in a pay-as-you-go approach, we call it a Public Cloud. The term Private Cloud is used to refer to internal data-centers of any organization which is not made available to the common public. A Hybrid Cloud is a combination of public as well as private clouds.

Cloud computing services hierarchy is topped by Cloud Robotics, a new paradigm and a *System of Systems* in itself [4].

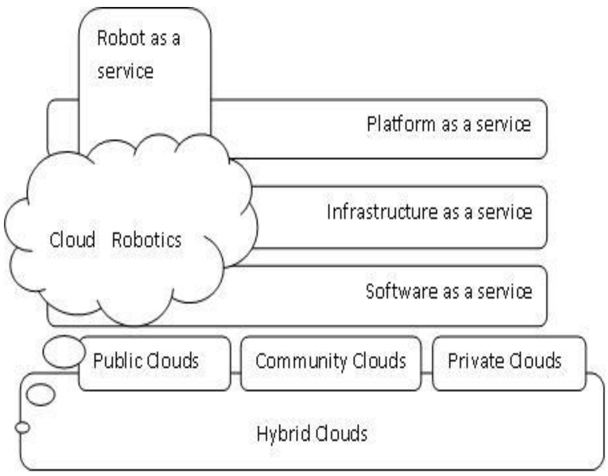

Fig 2: Cloud Computing Service Models, along with service layers and Robot as a service topped by Cloud Robotics [S. Jordan et al in "The rising prospects of cloud robotic applications"].

Cloud Computing provides 3 basic types of services:

Software as a Service (SaaS): In this model, software application is hosted as a service and end-users use the application on the browser.

Platform as a Service (PaaS): In this model, the end user creates, tests, and uploads applications using tools and libraries hosted by the service provider.

Infrastructure as a Service (IaaS): This model involves hosting of hardware computing services like storage, hard drive, servers and network components.

4. ROBOT OPERATING SYSTEM

Willow garage preserves ROS as an open source project [20] which offers a graph-like pattern for networked robotics. It was developed by Stanford in 2007 under the name Switchyard [2] originally. Robot Operating System (ROS) provides package tools and operating system-like tools [1].

Basically, as a collection of inter-programming language headers, ROS allows independent, individual programs, referred to in ROS nomenclature as nodes, to share data between themselves [10]. ROS is based on nodes, messages, topics, and services. In ROS, Nodes are the software modules or processes in the control code. Nodes communicate with each other through message passing i.e., passing multiple messages. A special type of message, called service, consists of a pair of messages, one for request and the other for reply. Nodes can publish and subscribe to a single (or multiple) topics.

Jonathan Bohren, in ROS Crash-Course Part I [6], mentioned that ROS is:

- Not an OS but a "meta" operating system for robots.
- A set of different tools for packaging and software building.
- Architecture for distributed intra-/inter-machine communication and configuration.
- System runtime and data analysis development tools.
- Under permissive BSD licenses, it is open-source (ROS core libraries).
- Has an architecture that is language-independent (C++, python, lisp, Java, and more).

- A platform that is scalable (ARM CPUs to Xeon Clusters).

He also mentioned that ROS is not:

- An actual operating system (OS).
- A programming language to write codes.
- A programming environment / Integrated Development Environment (IDE).
- A hard real-time architecture.

Jürgen Hess *et al* mentioned that there is more to ROS [7]:

- (De-)Serialization under the board.
- Central multi-level logging.
- Preemptible actions, service calls.
- Centralized time.

According to a site robohub.org [16], "ROS (Robot Operating System) is a BSD-licensed system for controlling robotic components from a PC." There are a number of independent nodes in any ROS system. Each node communicates with one another using a 'publish/subscribe' messaging model. For example, an infrared sensors' driver can be thought of as a node that publishes sensor data in a stream of messages. These messages could be used by several other nodes.

The developers at Google and Willow Garage have developed a rosjava library [14]. This new library is the first implementation of ROS using pure Java and was developed at Google with the goal of enabling sophisticated Android applications for cloud robotics. Since the entire library, tools, and the hardware that come along with Android devices are perfectly suited for robotics, Cloud Robotics with ROS can be a very good platform for the future of robotics development.

5. USING ROS IN CLOUD ROBOTICS

ROS can be used to create robots that can be used for our daily purposes and Cloud Robotics would help us augment the information contained in these robots. So, basically, we use ROS to develop robotic applications and CR to maintain them.

The Personal Robot (PR2) is a portable manipulation platform built by Willow Garage [14]. The entire (PR2) software structure is written with the help of ROS leading to which all PR2 capabilities are available to use via ROS interfaces. So, Cloud robotics using ROS is not too far from sight. In days to come, Robot operating system can be used to develop robotic applications that will work with the cloud in the cloud.

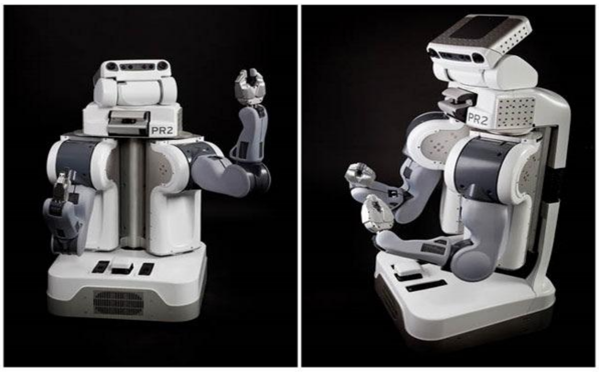

Fig 3: PR2 is a generic research platform widely used for cloud applications. The PR2 has demonstrated various capabilities relying on the Robot Operating System (ROS) [10].

We chose ROS over other Robotic middleware because ROS is much more easy to use and flexible and as per the needs of the user. Since it is an open-source platform for robot software development, it is all the more easy to maintain.

"The primary goal of ROS is to support code reuse in robotics research and development." - from www.ros.org [13].

Focusing on the primary goal of ROS, we can use it in cloud robotics so that code created for one robot can be easily reused in others too.

Aldebaran Nao [2] is a commercially existing humanoid robot that is around 60 cm tall and is targeted at research labs and classrooms. The Nao robot is small in size; still it has a lot to present even with its miniature frame. It has features like four microphones, multiple touch sensors on the robot's head, two VGA cameras, several infrared (IR) sensors, and many more.

The use of Nao [14] with the Robot Operating System has shown how quickly and easily open-source code can enable a community to come and work together around one common hardware platform.

Fig 4: Aldebaran Robotics NAO H25 Humanoid Robot [9].

6. FUTURE ASPECTS OF CLOUD ROBOTICS

Future robotic applications will benefit from cloud robotics, which provides the following advantages over traditional networked robots [19].

- Future Robots will have the ability to offload computation-intensive tasks to the cloud. The robotic platform becomes lighter and less expensive with easier to maintain hardware. To enable real-time actions, the robots only have to keep necessary sensors, actuators, and basic processing power. The maintenance of software onboard with the robots also becomes simpler, with less need for regular updates. The operational life and usefulness of the robotic network can be extended quite easily since the cloud hardware can be upgraded independently from the robotic network.

- Future Robots will have access to shared knowledge and new skills. A medium is provided by the cloud for the robots to share information, learn new skills, and acquire additional knowledge from each other. The cloud can host a database or library of skills or behaviors that map to different task requirements and environmental complexities.
- Future Robots will have access to large amounts of data. The robots can acquire knowledge to execute tasks through databases in the cloud. They do not have to deal with the creation and maintenance of data that comes in and goes out of the cloud.

7. CONCLUSION

ROS supports the philosophy of modular, tool-based software development in Robots. Even though ROS cannot be used as a framework for all Robotics software development, it can be extended by other software systems in future so that not only a specific set of challenges are met but every new challenge is considered and a solution is found. Cloud Robotics has a very essential challenge of determining the optimal ratio between online and offline computation. ROS can further be extended to create a balance in the online and offline computation done by robots.

8. REFERENCES

[1] Aaron Staranowicz, Gian Luca Mariottini, "A survey and comparison of commercial & open-source robotic simulator software".

[2] A. Bargar, "A comprehensive ROS interface for the Aldebaran NAO," August 2012.

[3] Ayssam Elkady and Tarek Sobh, "Robotics Middleware: A comprehensive literature survey and attribute-based bibliography", Jan 2012.

[4] E. Guizzo, "Robots with their heads in the clouds," IEEE Spectrum, 2011.

[5] Jit Ray Choudhary, "ROS: Robot Operating System."

[6] Jonathan Bohren, "Introduction to ROS distribution, build system and infrastructure."

[7] Jürgen Hess, Felix Endres, Armin Hornung, Bastian Steder, and Jürgen Storm, "ROS: Open Source Robot Operating system."

[8] L. Youseff, M. Butrico, D. Da Silva, "Toward a Unified Ontology of Cloud Computing. In Grid Computing Environments Workshop," 2008.

[9] NAO H25 | Humanoid Robot | Active Robots [online] Available: http://www.active-robots.com/aldebaran-robotics-nao-h25-humanoid-robot-academic-edition.html

[10] Parker James Conroy, "The Development Of An Aerial Robotics Laboratory Highlighting The First Experimental Validation Of Optimal Reciprocal Collision Avoidance," August 2013.

[11] Priyanki Jayantilal Vashi, "Cloud Robotics: An emerging research discipline."

[12] Robotic Evolution. ROS — the Artificial Intelligence for Robots. Available: http://robotsfuture.blogspot.hu/2011/07/ros-artificial-intelligence-for-robots.html, 2011.

[13] Robot Operating System [online] Available: http://www.ros.org.

[14] Robots using ROS: Aldebaran Nao [online] Available: http://www.ros.org/news/2010/03/robots-using-ros-aldebaran-nao.html

[15] Robots/PR2 – ROS Wiki [online] Available: http://wiki.ros.org/Robots/PR2.

[16] ROS 101: Intro to the Robot Operating System [online] Available: http://robohub.org/ros-101-intro-to-the-robot-operating-system.

[17] S. Jordán, T. Haidegger, L. Kovács, I. Felde, and I. Rudas, "The rising prospects of Cloud Robotic Applications," July 2013.

[18] Victor Chang, Robert John Walters, Gary Wills, "Review of Cloud Computing and Existing Frameworks for Cloud Adoption."

[19] Wee Peng Tay, Yonggang Wen, and Guoqiang Hu, "Cloud Robotics: Architecture, Challenges, and Applications."

[20] Willow Garage, "Robot operating system," 2010.

[21] Zhihui Du, Weiqiang Yang, Yinong Chen, Xin Sun, Xiaoying Wang, Chen Xu, "Design of a Robot Cloud Center."