

Toward Quantitative Emulation of Cloud Robotic Systems

Christopher C. Lamb, Rafael Figueroa, Rafael Fierro
University of New Mexico
Department of Electrical and Computer Engineering
Albuquerque, NM 87131-0001
{cclamb, rafa, rfierro}@unm.edu

ABSTRACT

Networks of collaborating autonomous cyber-physical systems are just beginning to be able to take advantage of cloud computing. As the advantages of using cloud computing become more obvious and validated in other fields, cloud computing will inevitably begin to migrate into autonomous system development. Our goal overall is to begin to provide quantitative guidance around how system developers can take advantage of cloud computing within their systems. In order to do this, we have embarked on two specific research thrusts; the first, building a cloud computing framework specifically tailored to the unique needs of autonomous systems, and the second, developing the appropriate simulation, emulation, and other tools needed to facilitate autonomous system cloud development. Our first step, outlined within this paper, is to develop an initial taxonomy of the types of distributed system architectures system developers will likely use, to notionally describe where and when they should be used, and how they could be developed and integrated.

Categories and Subject Descriptors

D.2.11 [Software]: Software Architectures—*Domain-specific Architectures*

General Terms

Design, Performance, Robotics

Keywords

Robotics, Cloud

1. INTRODUCTION

In the last two years, the robotics and autonomous systems community has begun looking closely at cloud computing to determine its efficacy in helping shoulder the computational load of systems. Overall, the consensus is that

cloud computing is positioned to make as large an impact in robotics over the next few years as it has in general Information Technology in the last. Projects that have begun to explore distributed knowledge management have demonstrated promising results [4, 5, 6], and have spawned other similar approaches in related domains.

We believe this approach to using cloud computing to help shoulder the computational burden for autonomous systems is not only promising, but here to stay, and also fraught with unique challenges. First, the economic benefits seem undeniable. Just as the cost and flexibility of cloud systems provided by commercial service providers like Amazon and Rackspace have encouraged an exodus of processing power from the company data center to the cloud, similar flexibility and cost savings will exist for robotics engineers. Today, in order to provision a network of robots, researchers need to acquire and administer laboratory computers, networks, and robotic systems prior to making any progress. Furthermore, these resources are centralized within a single laboratory and are usually inaccessible remotely. This leads to the very problem cloud computing set out to solve; an high capital-intensive barrier to entry and inconvenient resource centralization. Incorporating cloud computing into typical experimental environments makes processing power and storage more accessible than ever before, promising to bring new engineers into robotics as the Arduino brought programmers into the world of micro-controllers.

Not only can cloud computing lower barriers to entry, it can also more effectively provide processing power to networked autonomous systems. Instead of requiring local data storage and processing power, or planning systems that while remote are still centralized on a single workstation in a laboratory, the ability to take advantage of the virtually unlimited computational resources of cloud computing allows robots to run just about anywhere, distributing computation allows for resource intensive applications to be offloaded from robots to inexpensive cloud systems.

While the argument in favor of cloud computing for robotic systems mirrors that used with respect to traditional server-based computing, there are some distinct differences between the computation, storage, and communication needed by autonomous systems as opposed to simple servers or workstations. For example, communication latencies have a much larger impact on autonomous systems than non-mobile systems. An actuator managing a floodgate and controlled by cloud-integrated micro-controller afford to wait for direction from an external service under flood conditions. Likewise, a cloud-enabled grasping arm needs to be able to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICCPs '14, April 14-17, 2014, Berlin, Germany.

Copyright 2014 ACM 978-1-4503-1005-5/11/10 ...\$15.00.

quickly acquire grasping information from external sources whenever confronting a new object.

In order to effectively build these kinds of systems, we must first understand what kinds of systems we could build, under what circumstances, and why.

2. CYBER-PHYSICAL CLOUDS

In 2012, Hu, Ta, and Wen outlined specific differences between specific types of robotic communication in clouds [2]. Specifically, they rigorously described the differences between machine-to-machine and machine-to-cloud communications, and analysed optimum communication and computation distribution based on energy requirements. A valid and useful perspective, we intend to expand this to provide practical guidance with respect to how partition computation and data storage from an implementation perspective. We propose a structured way to look at cloud robotic systems based on the individual robot, the specific task, and the context in which the robot is executing that task. Based on this description, we intend to establish quantitative guidance and qualitative heuristics that help system designers determine how storage, computation, and communication can be distributed and dynamically tuned in operational systems.

First, we need to clearly define what we mean by *cloud robotics*. In order to clarify this frequently overloaded term, we turn to the US National Institute for Standards and Technology (NIST), the most authoritative group to yet propose a cloud computing definition. In Special Publication 800-145, Mell and Grance go into great detail describing clouds and different cloud types [3]. They define cloud computing as having five *essential characteristics*, three *service models*, and four *deployment models*. The service models include such logical functional partitions as Software as a Service, Infrastructure as a Service, and Platforms as a Service, while deployment models include various cloud types like public, private, hybrid, and community clouds. Service models certainly have an impact on the type of computing offered to robotic platforms, in that robots could consume exposed cloud functions in any of these service models. Interestingly, individual systems or constellations could be offered as a PaaS or IaaS offering to interested consumers as well, implying that as robots can take advantage of cloud computation, perhaps computational systems can take advantage of constellations of robots. In the scope of our current work however, we are most interested currently in the five essential characteristics of cloud computation:

- **On-demand Self Service** — Services can be easily provisioned without human intervention at a service provider.
- **Broad Network Access** — Services are ubiquitously available via computer networks.
- **Resource Pooling** — There is a certain level of geographic opacity to computational services, and services are pooled and dynamically reassigned according to need.
- **Rapid Elasticity** — Services can rapidly scale up or down according to need.
- **Measured Service** — Service providers monitor offered services in order to more effectively manage them.

These characteristics allow us to differentiate between *distributed* computation and *cloud* computation. For example, a fixed amount of data storage shared between members of a constellation would be distributed, while a virtually unlimited amount of storage shared between those same members would be a cloud service. Likewise, a Hadoop [1] implementation in which n nodes are reserved for each member would be distributed, while that same implementation with variable limits imposed by demand would be a cloud system.

Highlight the problem we are starting to address.

2.1 A Taxonomy of Architectures

Outline an initial taxonomy.

2.2 Initial Analysis of Taxonomy

Notionally analyze the taxonomy.

2.3 Future Work

Where to next.

3. RELATED WORK

Cover previous work in area, especially motivating work.

4. SUMMARY AND CONCLUSIONS

5. REFERENCES

- [1] 2013.
- [2] Hu, Tan, and Wen. 2012.
- [3] Mell and Grance. 2011.
- [4] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz. Web-enabled robots. *Robotics Automation Magazine, IEEE*, 18(2):58–68, 2011.
- [5] M. Tenorth, A. Perzylo, R. Lafrenz, and M. Beetz. The robotearth language: Representing and exchanging knowledge about actions, objects, and environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1284–1289, 2012.
- [6] M. Waibel, M. Beetz, J. Civera, R. D’Andrea, J. Elfring, D. Galvez-Lopez, K. Haussermann, R. Janssen, J. M. M. Montiel, A. Perzylo, B. Schiessle, M. Tenorth, O. Zweigle, and R. van de Molengraft. Robotearth. *Robotics Automation Magazine, IEEE*, 18(2):69–82, 2011.