q

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 as 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 [8, 9, 10], 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 set 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 Arudino 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 resoure 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

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

This presents challenges to those conducting research into cloud robotics however. In the past, research groups including ours has conducted experimental work using physical robotic systems mirroring those upon which we propose to execute developmental algorithms or deploy sensor platforms. This approach will not work when conducting cloud computing research. Researchers will not be able to afford the numbers of autonomous systems required to test approaches at realistic scale. As a result, we feel that it is vital that the community begin to develop verified and validated simulation and emulation tools that enable researchers to easily test approaches at simulated scale. We believe these tools must be able to run on a variety of systems, ranging from laptops for quick and dirty experimentation, to systems that incorporate physical resources like cloud nodes and robots and robotic constellations.

In order to effectively build these kinds of systems and the tools needed to emulate them, 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 [3]. 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 [7]. 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 scape 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.

Having established a baseline definition of the differences between distributed and cloud computing, we can begin to outline the various ways we can assemble cloud robotic systems.

2.1 Cloud Robotic Configurations

Computation services can be distributed throughout a constellation of autonomous systems. The primary services we are interested in are processing and storage. Communication is an important secondary service as cloud and distributed service acquisition and use, of any type, depends on communication capabilities. This gives us three service attributes we must model to understand how different configurations impact system design. Previous work has characterized services with respect to power consumption [3]. While vital to designing effective robotic systems, system designers will have other characteristics in mind as well, including system latency, storage and computational performance, and overall quality of service.

Generally, we can expect robotic networks to adhere to generally accepted networking patterns: stars, buses, rings, meshes, point-to-point communication, and trees. Of these generally accepted patterns, point-to-point communication degenerates into star networks with more than one participant, and can therefore be treated as a special case of a star network. Communication exists ubiquitously, between all nodes, while specific services can be distributed heterogeneously throughout the network. For example, all nodes may have some small amount of storage, but only specific dedicated nodes have extensive storage and the ability to manage information to in effect provide unlimited storage on demand. Processing power can be likewise distributed.

Furthermore, large networks of autonomous systems can be organized in hierarchical layers in a tree topology, similar to how the internet itself is organized. Specific cloud services can be distributed within any and all of these layers, depending on specific operational needs. In a typical

service topology, we may have an operational layer, consisting of various sensor platforms and actuators. That may be managed by a local layer, which is in turn managed by a regional and finally a centralized layer.

Figure 1: Service Distribution through Constellation.

Hu et. al. established two communication primitives for cloud robotic systems — machine-to-machine (m2m) and machine-to-cloud (m2c). Using these two primitives as reusable communication components, we can assemble them into a variety of different configurations.

Outline an initial taxonomy.

2.2 Initial Analysis of Taxonomy

Notionally analyze the taxonomy.

2.3 Future Work

We have defined how we can incorporate cloud computing into constellations of robots. In our proposed systems we have not only proposed external cloud computing services that can be used by various autonomous members of a given constellation, we have also shown how cloud computing capability can be distributed within a given constellation. The first common perspective on cloud computing has been addressed by others from a variety of different perspectives [6, 4, 2], the latter perspective on constellation embedded computing services is new and purports to have skewing effects on currently accepted communication and computation models addressing cloud robotics. At this point, we intend analyze our proposed models using Hu's power consumption effects model [4] as well as from other attributes of interest to system designers like latency and performance.

3. RELATED WORK

In 2012, Kamei et. al. proposed establishing a field of *Cloud Networked Robotics*. While not directly concerned with using cloud computing in the context of autonomous systems, Kamei and colleagues present a case supporting swarms of robots with distinct functions that collaborate to effectively support individuals. They do however describe the need to provide distributed computation and data storage services to swarms of robots enabling information sharing and coordinated planning. Key needs they identify for distributed computing systems include providing abstractions for individual robots, so a swarm can be automatically coordinate actions toward a goal rather than requiring programming for single entities, and the ability to share different types of information freely among cooperating robots [5]. Lorencik in 2013 outlined the state of the art in cloud systems supporting robotics deployments, outlining advantages to using cloud computing with robots in cost and performance domains, at the expense of network connection and latency dependence [6]. In 2012, Hu outlined both two primary patterns of communication within cloud robotics

as well as rigorous guidelines. based on power demands, on when they should be used and why. He also presented communication protocols useful in both scenarios, and presented communication performance expectations. Overall, this thorough evaluation characterizes computation based on power utilization well, but does not address other areas of concern, like urgency and result latency. Likewise, the structural communication patterns presented are well analyzed with respect to communication performance and information distribution, but those patterns are not yet addressed from other perspectives, like urgency, or data size, nor are specific uses motivated [4]. Chen proposed viewing robots as a service withing deployed constellations in 2011. In this model, which he did implement, he developed clear interfaces to robots using web-centric technologies in which robots could be controlled as a service rather than as a specific robotic entity. The implementation follows accepted service-oriented-architecture principles well, treating each robot as a service unit with specific services published within a larger service directory. These services can then be aggregated into specific applications within a given robotic constellation. Chen developed and implemented the original robot as a service model, but as of yet has not provided analytical guidance on when this kind of model is appropriate, exactly why it should be used, and how it could be implemented at scale in the presence of [2]. Quintas, in 2011, outlined and implemented a hybrid system of smart room sensors and robots integrated via message-based service oriented systems. This use of service oriented architecture (SOA) is closely related to cloud computing systems, and though it does not display the provisioning or scale of cloud computing in the outlined system, implementation of the services on top of a cloud provider's infrastructure would certainly address this gap. Overall, the proposed sytem is both well designed and reflects common SOA services including service discovery, and demonstrates the applicability of this general class of architectures to cloud robotics. It does not however provide any guidance with regard to when this kind of system should be used, or how it should enhance the cost or performance of integrated robotic systems outside of implementing the Trinity Scenario.

4. SUMMARY AND CONCLUSIONS

5. REFERENCES

[1] 2013.
[2] Y. Chen, Z. Du, and M. Garcia-Acosta. Robot as a service in cloud computing. In *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*, pages 151–158, 2010.
[3] Hu, Tan, and Wen. 2012.
[4] G. Hu, W. P. Tay, and Y. Wen. Cloud robotics: architecture, challenges and applications. *Network, IEEE*, 26(3):21–28, 2012.
[5] K. Kamei, S. Nishio, N. Hagita, and M. Sato. Cloud networked robotics. *Network, IEEE*, 26(3):28–34, 2012.
[6] D. Lorencik and P. Sincak. Cloud robotics: Current trends and possible use as a service. In *Applied Machine Intelligence and Informatics (SAMI), 2013 IEEE 11th International Symposium on*, pages 85–88, 2013.
[7] Mell and Grance. 2011.

[8] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz. Web-enabled robots. *Robotics Automation Magazine, IEEE*, 18(2):58–68, 2011.

[9] M. Tenorth, A. Perzylo, R. Lafrenz, and M. Beetz. The roboearth language: Representing and exchanging knowledge about actions, objects, and environments. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1284–1289, 2012.

[10] 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. Roboearth. *Robotics Automation Magazine, IEEE*, 18(2):69–82, 2011.