The Economic Case for Cloud-based Computation for Robot Motion Planning

Jeffrey Ichnowski, Jan Prins, and Ron Alterovitz

University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175, USA
{jeffi,prins,ron}@cs.unc.edu

Abstract. Imagine a world in which robots are a part of everyday life, performing elegant and safe motions to accomplish complex tasks. To achieve this vision, robots will need access to extensive computational resources. Cloud-based computers have the potential to provide the needed computing power, while lowering robot cost, space, and energy requirements. Academia and industry are already exploring the cloud as a purveyor of data in a wide variety of applications, and have shown the benefit of the cloud for accelerating offline- and pre-computations. But what about interactive/online computation, as is often required by robot motion planning? This paper presents an economics-based argument that it is possible to extends a robot's useful service life and battery operation time, improve its efficiency and profitability, and reduce its initial costs, by using the cloud in complex online and interactive computations. Gaining these benefits presents new, open research challenges, including: how to cost-effectively allocate cloud-based parallel computation, how to handle the unavoidable network-related bottlenecks, and how to design algorithms that distribute computation between the cloud and the robot.

Keywords: cloud computation, motion planning, parallel computation

1 Introduction

Consider a robot that needs to solve very challenging motion planning problems quickly. For high degree of freedom robots, a typical setup requires a capital expense of thousands of dollars to purchase a high-end computer capable of computing timely solutions. As an alternative, would you prefer gaining access to the latest computational hardware on demand, and for cents per task? That is the promise of cloud computing for robots—a potential to lower costs and improve efficiency for a variety of robotics applications.

Cloud computing has the potential to change the way we design, use, and pay for robotic systems. Unlike traditional robots which are purchased upfront, cloud computers are billed in units of usage time. Thus, when using cloud computing one can and should approach solving problems in the most cost-effective way possible. To illustrate, for \$10,000 one could purchase a high-end computer, or one could get 100,000 hours on a compute-optimized 1-core cloud computer, 6,285 hours on an 18-core cloud computer, or 1 hour of 113,136 cores[1]. With an embarrassingly parallel algorithm such as a sampling-based motion planner [1], one could dramatically reduce the time to solve a complex task. New robotics algorithms that leverage this computing power may extend a robot's service life and battery-based operation time, and reduce its initial and operating costs.

The cloud is already changing the way we think about computing for robots, but its full potential has not been tapped. To date, many data-centric, and pre-computation approaches leverage the cloud [11]. What about solving complex tasks with *near-term* deadlines by using the cloud to add computing power in response to the demands of a problem? This will be particularly valuable for network-connected robots that face challenging motion planning problems that involve high degree of freedom systems, dense cluttered environments, learning complex task models, or managing high levels of uncertainty. In this paper, we present an economic motivation for, and the research challenges posed by, leveraging cloud-based computation in online and interactive robotic algorithms. Bringing the benefits of cloud-based computing to robots poses multiple open research challenges, such as: how to cost-effectively allocate computing, how to design algorithms around network bottlenecks, and how to split computation between a robot and the cloud.

2 The Economic Potential of Robot Cloud Computing

The cloud changes the cost model of computing by shifting it from a capital expense (CapEx) to an operational expense (OpEx). Typically, robots require a large upfront CapEx, driven in part by the cost of the robot's computer. Using the cloud makes computing become an OpEx over the service life of the robot. With the right algorithms and utilization, an increase in a robot's OpEx will be offset by, not only a reduced CapEx, but also an increased service life, increased battery-based operation time, and a net improvement in operational efficiency.

Lower CapEx by extending a robot's service. A robot's service life may be extended through the use of cloud computing. The service life starts at purchase and ends when the robot's utility decreases to the point it is removed from service. Increasing the service life reduces the number of robots purchased over time, leading to a reduced CapEx. Consider a home assistance robot that aids someone with variety of daily tasks of living. Such a robot could gain additional functionality by following a process similar to that of installing applications and updates to a smartphone or tablet. In this scenario, the robot becomes obsolete and needs replacement due to either physical component wear or due to advances in software exceeding the capabilities of the robot's computing hardware.

Historically, computing hardware has become obsolete much more quickly than non-computing hardware (e.g., motors, sensors). Smartphones, as a proxy for a robot's computing platform, have a life expectancy in the range of 3 to

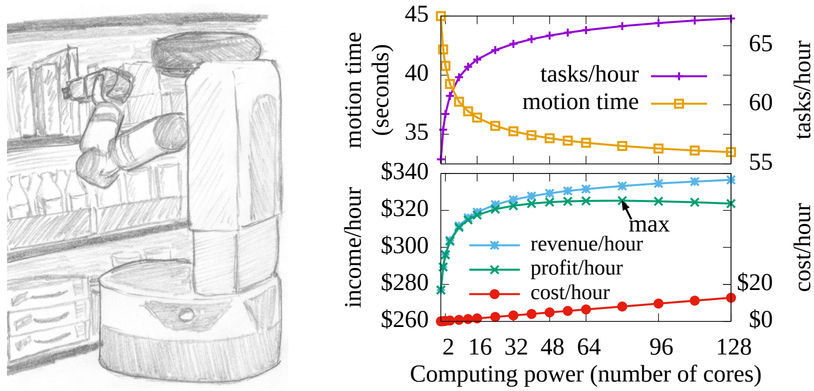
[1] As of June 2017, Amazon offers 1-core servers at $0.1/hr, and 18-core at $1.591/hr.

4.7 years [2,4]. Cars, as a proxy for a robot's non-computing hardware, have an average age in the US of 11.1 years [6]. The short service life of mobile computing devices is unsurprising when considering Moore's Law, which observes an 18-month doubling in computation power as measured by transistor count. At the end of a 4.7 year service life, a robot will have almost 9 times less computing power than its replacement. At the end of a car's 11.1 years, a robot will have almost 170 times less. The computing platform on a robot is fixed, but cloud services offer computers that are routinely upgraded. Thus, with appropriate algorithms that leverage cloud-based computation, a robot's service life would no longer be limited by its onboard computing hardware and it could operate until the non-computing hardware wears out, potentially adding years to the robot's service life.

Cheaper robots with longer battery life. Incorporating a reliance on cloud computing into the physical design of a mobile robot will allow for cheaper robots with longer battery-based operation time. The computing platform in a robot is necessarily limited by economic factors, including price and, for mobile robots, physical size and battery capacity. Embedding high-end CPUs and GPUs enables higher performance computing, but comes at a cost of dramatically increased price and energy drain for the robot. Higher energy drain either requires increased battery size and weight, or results in reduced battery-based operation time. If instead, a robot's designers look to lower-power computing platforms sufficient to running baseline algorithms, while offloading intensive computation tasks to the cloud, their robot design can offer decreased battery size or allow for an increased battery-based operation time, all for a reduced upfront cost.

Robots that learn from their environment and from humans are examples that could naturally benefit from such a cloud-enabled robot design [14]. Cloud-based computation accelerates the learning of a model, while the robot need only use the learned model with low-powered computation. Such a system could be used in robots that are deployed to unfamiliar environments and expected to adapt rapidly to them as they operate. Novel cloud-based learning solutions [9], and low-powered fast convolutional network processors [5] and FPGAs [13], are making this closer to reality.

Improved operational efficiency. Cloud computation can not only reduce the initial purchase costs of a robot, but it can also increase a robot's operational efficiency, potentially increasing associated revenue and reducing the need to purchase more robots. Motion planning can be computationally intensive, whether attempting to find a feasible solution in a complex space, maximizing a task's success rate in the presence of uncertainty, or minimizing a motion's cost (e.g., path length, time to completion, energy required). Asymptotically optimal [10] and near-optimal [12] motion planners work to minimize a motion's cost by converging towards optimality. They converge faster when given more computing power or computational parallelism [7]. By leveraging parallelism of cloud computers for motion planning [3,8], robots can complete tasks faster.



(a) Robot picking from a shelf (b) Efficiency gained with cloud computers

Fig. 1: Example of maximizing profit with cloud computing. A warehouse packs packages using robots (a). The robots avoid collision with an ever-changing inventory by using motion planning algorithms. In (b), the warehouse wishes to maximize *profit/hour*, which here is computed as $(revenue/hour) - (cost/hour)$. Each task the robot completes results in revenue for the company, thus more *tasks/hour* means more *revenue/hour*. The robot uses cloud computation of an asymptotically optimal motion planner to reduce *motion time* and thus pack more boxes per hour. When the cost of adding more computing outweighs the gain in revenue due to higher quality motion plans, profit is maximized.

When accelerating robot motions results in more revenue, the OpEx associated with cloud computing could be justified by net improved profits (see Fig. 1). When a fixed number of tasks are required per unit time, faster completion times means fewer robots are required, thus lowering CapEx.

3 Research Challenges

Cloud computing offers many potential benefits, but realizing them presents several open research problems. Cloud computing services offer the scaling of computing power and a wide variety of configurations, from a single core virtualized on a server, to all cores on a high-end multi-core computer, to arrays of GPUs, to networked combinations of these. Motion planning algorithms that benefit from parallelism typically run with fixed parallelism configured a priori. With cloud computing, the amount of parallelism to allocate to a problem becomes a question of balancing benefit to the cost (instead of availability) of computing. Robots also must interact with a changing world, and in order to respond to changes (e.g., to sense and avoid collisions with obstacles) they must take into account the network latency (i.e., round-trip time) and bandwidth limits. One option is mixing or splitting computing between multiple sites: the robot's onboard and the cloud-based computers, ideally gaining the benefits of each site's strengths while avoiding the weaknesses. The research challenges are thus: how to allocate parallelism afforded by cloud computing, how to model the costs of it, and how to adapt algorithms to work around limitations of the network.

Allocating computing resources. How does one best allocate parallel computing power when implementing an algorithm on a robot? That is the question often left as an engineering exercise to the reader of parallel computing research. The answer is typically: allocate as much computing power as you have available—e.g., all cores on a computer, all warps on a GPU, and/or all computers in a cluster. But with cloud computing, "as much computing power as you have available" is insufficient as an answer—since the amount of computing power available is typically beyond the financial limits of reasonability. This suggests the benefit associated with the amount of computing power must be balanced against the costs associated with using that computing power. For example, increasing the motion efficiency of a warehouse robot will have an associated benefit to the bottom-line of the company (money saved), so at what level of computing parallelism (money expended) does the company maximize profit? Thus the question of how to best allocate computing power depends on modeling economic costs. And while the exact answer is still application specific, as researchers we can look to supplying the tools to make the decisions. Such tools might include models of convergence rates and speedup associated with parallelism, and determining minimum or expected computational requirements associated with tasks.

Network bottlenecks and deadlines. Robot algorithms that rely on cloud computing must consider and address the limitations imposed by the network. Advances in networking technology may improve the latency and bandwidth to an extent, but communication networks will always be slower than the interconnect between the robot and its onboard or co-located computer. This limit is fundamentally insurmountable, since it is a direct result of the speed of light. As such, network limitations vary by domain, and the challenges imposed by the network bottlenecks for robots in home and warehouse environments significantly differ from robots tasked with deep-sea and space exploration. For the class of algorithms and scenarios in which the results can be pre-computed, the network might not warrant concern. However, robots operate in the real world, and they must be able to sense and respond quickly to changes in the environment in order to avoid undesirable or harmful outcomes, especially in safety-critical scenarios, such as warehouse robots operating in close proximity to humans or with medical robots working with, or operating on, humans. To avoid undesirable outcomes, we pose the research challenge by borrowing language from the real-time computing community, and considering computing tasks with *hard deadlines* and *soft deadlines*. For robotic tasks with hard deadlines (ones that cannot be missed), how can we ensure that a robotic algorithm will meet the deadline (or at worst minimize the chance of missing the deadline)? For robotic tasks with soft deadlines (ones for which a miss results in reduced benefit or increased cost), how can a robotic algorithm maximize the benefit or minimize the cost of these tasks? More parallel processing can speed up computation to get ahead of the deadline, but the network remains a bottleneck of fundamental importance to these research challenges.

Splitting computation between multiple sites. Algorithms can potentially address the resource allocation and network concerns by splitting computation between the robot's on-board computer, a co-located computer, and cloud-based computers. The robot's computer has the lowest latency and highest bandwidth access to its environment via its sensors and actuators. A cloud-based computer has relatively high latency and low bandwidth. Depending on the scenario in which the robot operates, some portion of the robot's computation can be split between the different sites. As an example, vision processing and motion tracking require a large amount of bandwidth and low latency in order to react to changes in the environment—matching the characteristics and (hopefully) capabilities of the robot's onboard or co-located computer. On the other hand, an intensive pre-computation of a robot's path through its environment (barring dynamic obstacles) can be rapidly computed by high-performance parallel computing in the cloud. As a general research challenge, can we design robot algorithms that split portions of computation between multiple computing sites and thus gain the benefits of the cloud's massive computing power while meeting the demands of a problem that requires low-latency computation?

4 Conclusion

Cloud computing has a great potential to benefit robot motion planning and related computationally intensive problems in robotics. New algorithms that leverage cloud computing can increase a robot's service life and improve its operational efficiency. Incorporating cloud computing into a robot's design can decrease initial costs while reducing size and weight and extending battery-based operation time. These potential gains motivate new open areas of research in cloud computing for robotics, including how to address the network bottlenecks, exploit the cloud's computing parallelism, model the costs of cloud-based robotics algorithms, and meet task deadlines. Finally, while our examples focused on motion planning as a particularly computationally intensive problem, we hope to inspire the robotics community to apply cloud computing to entirely new aspects of research not considered here.

Acknowledgement

We thank Jonathan Lynn for useful discussions regarding economic models. This research was supported in part by the U.S. National Science Foundation (NSF) under awards CCF-1533844 and IIS-1149965 and the U.S. National Institutes of Health (NIH) under award R01EB024864.

References

1. Amato, N.M., Dale, L.K.: Probabilistic roadmap methods are embarrassingly parallel. In: Proc. IEEE International Conference Robotics and Automation. pp. 688–694 (1999)
2. Apple: Environment - Answers, `http://web.archive.org/web/20170612120817/https://www.apple.com/environment/answers/`, accessed 2017-06-12
3. Bekris, K., Shome, R., Krontiris, A., Dobson, A.: Cloud automation: Precomputing roadmaps for flexible manipulation. IEEE Robotics & Automation 22(2), 41–50 (2015)
4. Ely, C.: The life expectancy of electronics (Sept 2014), `https://www.cta.tech/News/Articles/2014/September/The-Life-Expectancy-of-Electronics.aspc`
5. Esser, S.K., Merolla, P.A., Arthur, J.V., Cassidy, A.S., Appuswamy, R., Andreopoulos, A., Berg, D.J., McKinstry, J.L., Melano, T., Barch, D.R., et al.: Convolutional networks for fast, energy-efficient neuromorphic computing. Proceedings of the National Academy of Sciences pp. 11441–11446 (2016)
6. Ford, D.: As cars are kept longer, 200,000 is new 100,000. New York Times (Mar 2012), `http://www.nytimes.com/2012/03/18/automobiles/as-cars-are-kept-longer-200000-is-new-100000.html`
7. Ichnowski, J., Alterovitz, R.: Scalable multicore motion planning using lock-free concurrency. IEEE Transactions on Robotics 30(5), 1123–1136 (2014)
8. Ichnowski, J., Prins, J., Alterovitz, R.: Cloud-based motion plan computation for power-constrained robots. In: Algorthmic Foundations of Robotics (Proc. WAFR 2016). Springer (2016)
9. Jouppi, N.P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al.: In-datacenter performance analysis of a tensor processing unit. 44th International Symposium on Computer Architecture (ISCA) (to appear) (2017), preprint arXiv:1704.04760
10. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int. J. Robotics Research 30(7), 846–894 (Jun 2011)
11. Kehoe, B., Patil, S., Abbeel, P., Goldberg, K.: A survey of research on cloud robotics and automation. IEEE Transactions on Automation Science and Engineering 12(2), 398–409 (2015)
12. Marble, J.D., Bekris, K.E.: Asymptotically near optimal planning with probabilistic roadmap spanners. IEEE Transactions on Robotics 29(2), 432–444 (2013)
13. Murray, S., Floyd-Jones, W., Qi, Y., Sorin, D.J., Konidaris, G.: Robot motion planning on a chip. In: Robotics: Science and Systems (2016)
14. Waibel, M., Beetz, M., Civera, J., d'Andrea, R., Elfring, J., Galvez-Lopez, D., Häussermann, K., Janssen, R., Montiel, J., Perzylo, A., et al.: Roboearth. IEEE Robotics & Automation Magazine 18(2), 69–82 (2011)