Resource Utilization Prediction Model for SLAM Offload to Edge

KOKI NAGAHAMA† YOICHI ISHIWATA ‡ MIDORI SUGAYA†

Abstract: In recent years, the use of autonomous mobile robots has been increasing. SLAM technology is widely used for autonomous mobile robots to estimate their position and to create an environmental map at the same time. However, SLAM has a disadvantage because a load of various image processing and computation tasks is high. Also, in systems with extremely limited hardware resources, such as embedded systems, the execution speed is likely to be slowed down. Although various offloading methods have been proposed, they have not been applied to ROS+SLAM applications, and there are few examples of verification with robots. Therefore, in this study, we build a prediction model based on system resource information for a certain period associated with SLAM processing. Based on this resource usage prediction, we also propose a load-reducing method to offload data that is less frequently accessed on memory to the edges as needed. We construct a prediction model using the data obtained from the robot and describe the results of the evaluation.

Keywords: Memory prediction model, Offload, SLAM, ROS, Applications, Resource usage model

1. Introduction

In recent years, the use of autonomous mobile robots has been promoted in Japan as "the world's most robotic society"[1].

SLAM (Simultaneous Localization and Mapping) technology has been widely used in autonomous mobile robots to estimate their position and create an environmental map simultaneously [2]. SLAM is an indispensable technology for autonomous mobile systems because it enables autonomous robots to grasp their position and information about their surroundings. However, SLAM is computationally demanding because it requires physical information from several sensors, and various computational processes are required. This has led to concerns about a significant reduction in execution speed in systems with extremely limited hardware resources [3].

To reduce the computational burden of SLAM, Benjamin Sugar et al. proposed a method for optimizing and approximating least-squares maps by hierarchically subdividing and generating many sub-maps of small size with restricted dependencies by applying the divide-and-conquer principle [3]. Also, Alexander Schiotka et al. used a Monte Carlo localization method[4] to achieve a similar goal. These experiments in a real environment have shown that these approaches contribute to reducing memory usage.

These approaches achieved a reduction in memory consumption, mainly by improving the map generation algorithm in SLAM. However, since only specific SLAM algorithms are considered, only specific SLAM application improvements can be realized, which makes it difficult to achieve general versatility.

Therefore, in this study, we consider offloading to the edge, which is a close-range calculator, as a versatile way to reduce SLAM load on a computer with limited hardware resources. Edge is a distributed server with 5G wireless facilities that can be placed close to users and sensor nodes, etc., to receive advanced computing requests from IoT devices, etc. However, methods to effectively reduce the load on SLAMs using this technology have not yet been fully studied.

The purpose of this study is to investigate a method of offloading data to the edge using SLAM processing. We also propose a method to offload data that is not frequently accessed on the memory to the edge. In addition, in the case of real-time offloading of infrequently accessed data on the memory to the edge server, the timing of the transfer and the selection of data shall be optimized so that the processing itself does not become burdensome. To achieve these goals, we have developed a predictive model of memory consumption and network load for SLAM processing and discussed its feasibility for offload processing.

2. Proposal

2.1 Overview

It is important not only to determine whether offloading is necessary for SLAM based on load-related information to achieve generalized load reduction in SLAM but also to avoid the offloading process itself to be a load. In order to meet these requirements, it was necessary to create a predictive model of the different resource usage situations in each system, select the data that need to be offloaded, and decide the timing of the offload based on the predictions. For the prediction model, since memory and network are important as computational resources, we propose two model equations to achieve the optimal offload: (1) memory consumption prediction model equation and (2) network load prediction model equation.

2.2 Memory consumption prediction model

To predict the memory consumption in SLAM, we define a memory prediction equation. First, in order to predict memory, we need to know the amount of data in heap, stack, buffer and cache memory.

The definition of the prediction model for memory usage ratio is given by equation (1).

$$M = a_1 \times M_{stack} + a_2 \times M_{heap} + a_3 \times M_{buf} + a_4 \times M_{cache} \cdots (1)$$

$a_1, \cdots, a_4$ are the partial regression coefficients and M_{heap}, M_{stack}, M_{buf}, M_{cache} are the memory types shown earlier, respectively. In addition, the multiple regression model is important in making predictions because it has the advantage of minimizing errors.

2.3 Network load prediction model

The definition of the prediction model for network load is given by equation (2).

$$N_{load} = b_1 \times (N_{send} - \overline{N_{send}}) - b_2 \times (N_{receive} - \overline{N_{receive}}) \cdots (2)$$

Note that b_1, b_2 is a coefficient, which is determined by equations (3) and (4), depending on the acquired data.

$$b_1 = (N_{send} + N_{receive}) \times \frac{1}{\frac{N_{send}}{min(N_{send}, N_{receive})}} \cdots (3)$$

$$b_2 = (N_{send} + N_{receive}) \times \frac{1}{\frac{N_{receive}}{min(N_{send}, N_{receive})}} \cdots (4)$$

N_{load} is the network load level, N_{send} and $N_{receive}$ are the amount of data sent and received respectively, and $\overline{N_{send}}$ and $\overline{N_{receive}}$ are the averages of each of the above items. By using the result of this equation as an indicator of how much load is on

† Department of Information Science and Engineering, Faculty of Engineering, Shibaura Institute of Technology, Tokyo, Japan

‡ Ales Inc, Tokyo, Japan

the network, we believe that it has a significant impact on the decision of when to execute the offloading process.

3. Evaluation system

3.1 Evaluation system

We prepared one TurtleBot3, which has 1GB RAM, and a laptop that has enough large capacity memory and built ROS (Robot Operating System) for the OS installed in them. Data was acquired by executing SLAM processing in this environment. Before starting data collection, the memory load was increased by 0%, 25%, and 50%, and the acquired data were saved as datasets 1-3 in order of descending load. The first 80% of the dataset was used for training and the rest of the dataset was used as test data for evaluation, using 180 consecutive seconds in each dataset.

3.2 Memory consumption prediction model: evaluation methods and Result

For the evaluation of the memory consumption prediction model, the regression equation is calculated for each dataset by using multiple regression analysis, and the actual values are compared with the predicted values of the proposed model.

The results of the evaluation are shown in Figure 4 and Tables 3 and 4. From the graph in Fig. 4 and Table 3, the predicted values close to the actual values can be obtained, and the error is suppressed to within 2%.However, looking at the Multiple determination factor, the larger the load, the smaller the coefficient of determination, and looking at Table 4, the p-value exceeds 5%. So, it cannot be said that the accuracy is high. Therefore, improvement of the prediction formula is considered as an issue.

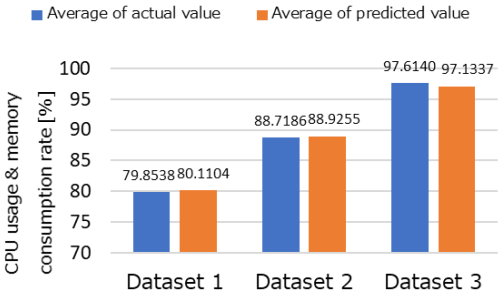

Fig. 4 Comparison of the average of actual and predicted value

Table 3 Multiple determination factor, F-value, and maximum error rate for each dataset

Dataset	Multiple determination factor	Significance F*	Maximum error rate
1	0. 7820	F < 0. 000001	1. 17%
2	0. 8097	F < 0. 000001	1. 70%
3	0. 5012	F < 0. 000001	0. 57%

Table 4 P-value of each element in each dataset

(** : p<0. 01, *: p<0. 05)

Dataset	P-value			
	buffer	cache	heap	stack
1	0. 0062**	p<0. 0001**	p<0. 0001**	0. 6158
2	0. 1660	p<0. 0001**	p<0. 0001**	0. 2040
3	0. 3042	0. 2764	p<0. 0001**	0. 4358

3.3 Network load prediction model: evaluation method

To evaluate the network load prediction model, coefficients b_1, b_2, are calculated for each dataset using equations (3) and (4), and the results are given into equation (2). For evaluation and discussion, we use the upper and lower limits of N_{load}, the probability of N_{load} going below 0 (the probability of occurrence of the timing when the load of network transmission is small, $p(N_{load} \leq 0)$, and the duration (maximum duration) of N_{load} as quantitative factors.

3.4 Network load prediction model: evaluation

The results of the evaluation are shown in Table 5.

For $p(N_{load} \leq 0)$, we estimate the load of the network communication using the model and find that there is a timing at least once every 3.3 seconds on average at which the impact of network transmissions can be determined to be small. Looking at the maximum duration, there is a possibility that network transmissions can be used for offloading for at least one second, even if the processing hangs due to high load on the upper memory. Considering the bandwidth of the network, the amount of data that can be transferred offload is sufficiently small.Therefore, we believe that N_{load} is good enough as a factor to determine the timing for optimal memory offloading.

Table 5 Upper/lower limits, $p(N_{load} \leq 0)$, and maximum duration for each dataset

Dataset	Upper/lower limits of N_{load}	$p(N_{load} \leq 0)$	Maximum duration[s]
1	19. 29 / -64. 10	33. 33%	2
2	24. 09 / -13. 48	14. 29%	2
3	0. 52 / -1. 55	5. 56%	1

4. Conclusion

The proposed multiple regression model for memory consumption and network load prediction model was constructed and evaluated based on the actual data collected during SLAM execution. This suggests that these models are sufficiently effective in predicting the memory consumption associated with memory offloading as well as in determining the timing. On the other hand, since the accuracy of the multiple regression prediction models for memory consumption varies, it is necessary to reconsider the components and improve the prediction accuracy.

In the future, we will study methods for solving the above-mentioned issues. In addition, we are planning to design and implement a memory offloading system that combines ROS Publish/Subscribe network and KVS (Key-Value Store), which is a data management system, and to evaluate the effectiveness of the system.

References

[1] Ministry of Education, Culture, Sports, Science and Technology Japan, Ministry of Economy, Trade and Industry. Toward the realization of a universal future social experience with advanced robot technology. 2015 edition, 2015. https://www.Kantei.Go.jp/jp/singi/keizaisaisei/wg/kaikaku/dai6/siryou3. pdf

[2] The Mathworks Inc., What Is Slam? 3 things you need to know. https://www. mathworks. com/discovery/slam. html

[3] Benjamin Sugar. An approach to solving large-scale SLAM problems with a small memory footprint. 2014 IEEE ICRA, 2014. p. 3632-3637

[4] Alexander Schiotka. Robot localization with sparse scan-based maps. 2017 IEEE/RSJ IROS, 2017. p. 642-647