Parallel Risk Management Framework for Maritime Domain Awareness

Alexander Teske
Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada K1N 6N5
atesk062@uottawa.ca

December 15, 2016

Abstract

A Risk Management Framework is augmented with the ability to distribute risk calculations across multiple CPUs. We demonstrate that a serial implementation of these calculations can become unacceptably slow for large input sizes. Our experimental results show that our approach achieves linear speedup with respect to the number of processors. The approach is applied to Maritime Domain Awareness as a case study.

1 Introduction

The Risk Management Framework (RMF) is a modular system for managing risk in generic distributed systems (e.g sensor networks). The two most integral RMF modules are "risk extraction" and "risk assessment". The former module receives raw data from all sensors and assigns each sensor one or more risk values (i.e. numbers between [0,1] where 0=low risk and 1=high risk) corresponding to different risk factors in the environment. The latter combines the risk values for each sensor, resulting in an "overall risk value" for each sensor. A concern is that these operations may become expensive in domains where there are many sensors. Since the RMF is intended to provide risk analysis in real time, this can be a real hindrance.

An example of a domain where the RMF can be applied is maritime domain awareness (MDA), where the task to be accomplished by the RMF is maritime risk assessment. Here the sensors are vessels at sea, which broadcast data including their position, speed, heading, etc. Risk factors in this domain range from possibility of two vessels colliding to the possibility of hostile actions against a vessel. Given the massive number of vessels which can potentially be at sea at any given time, a concern is that the "risk extraction" and "risk assessment" steps can become a bottleneck for the RMF.

The objective of this research is to determine if there is any value in parallelizing the "risk extraction" and "risk assessment" steps of the RMF. In particular, this would be the case if (a) the incoming data has sufficient velocity that the RMF cannot compute results in real time, and (b) the overhead of applying parallel computing does not outweigh the benefits. It stands to reason that condition (a) is domain dependant. It is more likely to be satisfied in domains that have many sensors and in domains where sensors send data very frequently. It is especially true when the risk factors of one sensor depend on the state of the other sensors. In this situation, an update from one sensor triggers the risk assessment for every sensor in the network.

In the maritime domain, the total number of vessels at sea is roughly one hundred thousand. In practice, the RMF will likely only track a subset of these. Furthermore, the collision risk (i.e. the risk of a vessel colliding with another) vessel depends on the position of all vessels, thus an update from one vessel triggers risk assessment for each other vessel. Therefore, the maritime domain seems like a promising candidate to benefit from parallel computing.

For this study, we have chosen a threshold of one second to be the maximum processing time that will be considered real-time. This means that if the execution time is less than 1s for a certain number of vessels, there is no incentive to use parallel computing. We note that this threshold is chosen arbitrarily since regardless of the chosen value we expect that there is a sufficiently large input that the data cannot be processed on time.

To the best of our knowledge there have been no efforts to parallelize calculations in maritime domain awareness, nor have there been any efforts to parallelize the RMF.

The remainder of this manuscript is structured as follows: Section 2 reviews the recent literature concerning Maritime Domain Awareness and the Risk Management Framework, Section 3 describes our approach to parallelizing the RMF's risk evaluation steps, Section 4 gives our experimental results, and we give our conclusions in Section 5.

2 Literature Review

2.1 Maritime Domain Awareness and Maritime Risk Assessment

Maritime Domain Awareness (MDA) is defined as the situational understanding of activities that impact maritime security, safety, economy or environment. One goal of MDA is to effectively coordinate assets to respond to illegal activities, disaster situations, and rescue scenarios in the maritime domain [1].

Maritime risk assessment is a task within MDA that involves monitoring and managing risk, for example the risk of two vessels colliding. There are many potential data sources which can be inputted to maritime risk assessment systems. Typically, these sources are classified as hard vs. soft. Hard data tends to be reliable, with a high sampling rate and good precision. Examples of this include radar data and automatic identification system (AIS) data. Soft data is usually considered less reliable. It may have an infrequent sampling rate and can be less precise. Examples of this include human recorded data such as field reports or text mining data.

To be effective, a maritime risk assessment system must be able to ingest large volumes of such data and produce outputs in real-time. The system must deal with data sources with high volume, velocity, veracity, and variety. Thus, maritime risk assessment can be considered a big data problem.

Many techniques have been put forward to handle the big data problem introduced by MDA. Hidden Markov Models (HMM) [7] [9] have been proposed to monitor risk in networks. However, [3] points out that HMMs but can be unstable in dynamic situations such as MDA. In particular, if several interconnected HMMs are used in maritime monitoring, a single change in the environment would require all models to be updated. This quickly becomes prohibitively expensive.

Recently, [8] used genetic programming and linear scaling along with AIS data to perform vessel path prediction. This approach was shown to outperform two different versions of genetic programming as well as three non-evolutionary algorithms.

[10] applied Bayesian belief networks to assess risk for vessels in the approach channel of the Tianjin port. This work identified areas where traffic was being managed inefficiently. However, the approach was tested on a dataset of only 234 collision reports. Furthermore, the timeliness of the calculation was not reported.

In [6], artificial neural networks were used to combine data from multiple optical sensors (e.g. visual, thermal, multi-spectral) to classify maritime targets in near real-time.

2.2 Risk Management Framework

The Risk Management Framework (RMF) is a modular system for managing risk within generic distributed systems in real-time. It was first proposed by Falcon et all in [5]. The primary functions of the RMF are (1) to assess the risk level for system units using data reported from the units themselves, (2) visualizing the risk landscape for human operators, and (3) generating potential responses to mitigate risk in the environment. In essence, the potential responses create a closed loop between the RMF and the environment; once the response is enacted upon the environment, the risk landscape changes and the risk values are reassessed.

In [2] the RMF was applied to the maritime domain. Here the system units are vessels at sea, and risk factors such as risk of collision with another vessel and regional hostility are calculated using information such as each ship's position, heading, and speed. When a vessel's risk level exceeds an acceptable threshold, the vessel is deemed a *vessel in distress* (VID). Vessels in distress must assisted in what is called a search and rescue (SAR) mission. This mission involves one or more vessels in the vicinity of the VID coming to the aid of the distressed asset. The RMF is capable of identifying VIDs and generating potential SAR missions using the remaining maritime assets. The potential responses are ranked according to several competing objective functions and the best ones are presented to a human operator.

In [4] the RMF, as applied to the maritime domain, was augmented with the ability to ingest soft data such as textual reports of maritime incidents. It was shown that the approach effectively converted soft information into quantitative data which could be used to evaluate risk.

3 Algorithmic Approach

3.1 Data Preprocessing

In the maritime domain, the raw data must be preprocessed prior to being submitted to the "risk extraction" step. In particular, the GPS coordinates of each ship are used to determine, for each vessel, the distance to the nearest other vessel.

The naive approach to this step would be to compute the distance between each pair of ships. This would require $\mathcal{O}(n^2)$ time and $\mathcal{O}(n^2)$ memory. With parallel computing, we could reduce this to $\mathcal{O}(n^2/p)$ time where p is the number of processors.

A more sophisticated approach would be insert the GPS coordinates into a space-partitioning data structure such as a kd-tree, quadtree, or a Voronoi diagram. These structures can generally be constructed in linear time, use linear memory, and allow nearest neighbour search to be conducted in logarithmic time.

Unless $n \approx p$ the space-partitioning data structure approach is superior to the naive approach. This holds true for this research, since $p < 32$ and $n > 100$. On the other hand, the complexity of building and querying the space-partitioning data structure does not depend on the number of processors. This means that the cost of the preprocessing step does not depend on whether the RMF is running in serial or in parallel. For this reason, this research used a random variable for the distance to nearest ship in order to simplify the implementation. The preprocessing step was therefore excluded from the benchmarking.

3.2 Parallel Risk Evaluation

The main focus of this study is to parallelize the risk evaluation step of the Risk Management Framework (RMF). This process involves using the RMF's "risk extraction" module to generate several risk values for each sensor, then using the "risk assessment" module to combine the risk values into one "overall risk value" for each sensor. Once the data preprocessing is complete (see Section 3.2) this task can be completed independently for each sensor.

Therefore, the following approach is put forward to parallelize the RMF: divide all risk evaluation calculations evenly among the available processors. If there are n processors and p sensors, each processor performs the "risk extraction" and "risk assessment" steps for roughly n/p sensors.

To accomplish this, the processor with rank 0 is assigned the role of master and all other nodes are workers. The master node reads raw data and maintains a list of all vessels along with their most recent data packet. Each time an update is received from a sensor, the master node assigns a subset of the vessels to each worker. The workers perform risk evaluation for their vessels, and report the overall risk values back to the master.

We expect this approach to provide linear speedup of the risk evaluation steps.

4 Experimental Setup and Results

4.1 Experimental Setup

The Risk Management Framework was implemented in the C++ programming language, and OpenMPI was used to distribute the computations on the cluster. Testing was performed on Carleton University's OpenStack cluster. A set of 32 instances was created for benchmarking purposes. Each instance had 4GB RAM and a single core processor clocked at 2.2GHZ. Each test was run 10 times and the average result was reported.

4.2 Experimental Results

Figure 1 shows the effect of increasing the number of vessels n on the execution time. For small n, the processing time is negligible. For example, at $n = 100$, the processing time is 2ms. Thus for small n there is no advantage to using multiple CPUs. However, the processing time increases linearly with n. For example, with $n = 3800$, the execution time is roughly 1s.

Since the execution time grows linearly with n, we can conclude that in any domain there is a certain value of n for which the RMF cannot compute results in real time. For maritime risk assessment that value is $n \approx 3800$, assuming that results must be returned in less than one second.

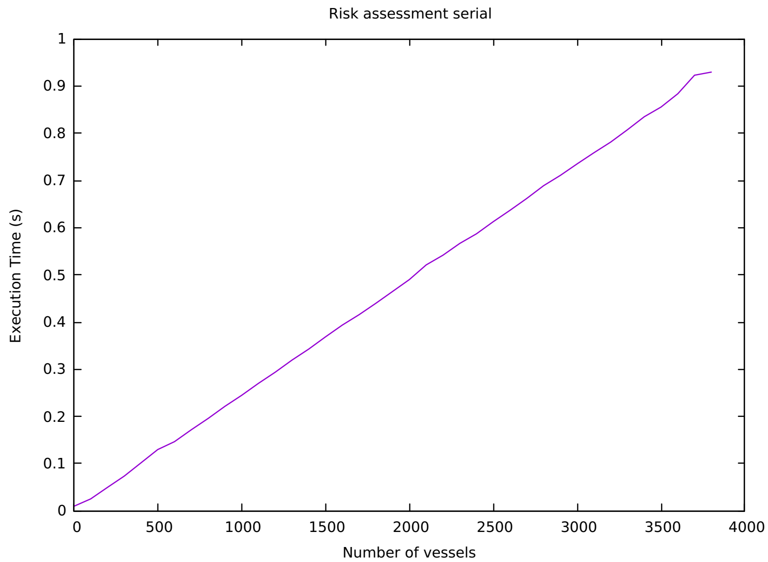

Figure 1: Risk Assessment On A Single CPU

Figures 2, 3, and 4 show the effect of increasing the number of processors from 2 to 32 with a constant input size.

In Figure 2, the input size is 100. The processing time decreases linearly with respect to the number of slave nodes until p=10, at which point performance begins to degrade as the overhead outweighs the benefits of adding CPUs.

In Figure 3, the input size is 1000. Here we see roughly linear speedup with the number of processors. However, we note that the speedup curve is not smooth since the input is not large enough to fully benefit from parallel computing.

Finally, Figure 4 show the results with input an input size of 5000. Note that this is the only test with an input size greater than the previously discussed threshold of $n = 3800$. At this input size, we very clearly see linear speedup.

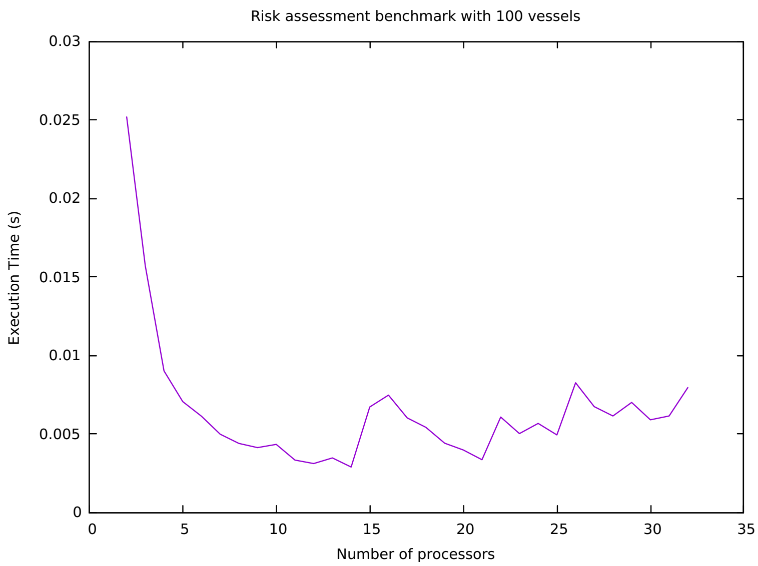

Figure 2: Risk Assessment for 100 vessels

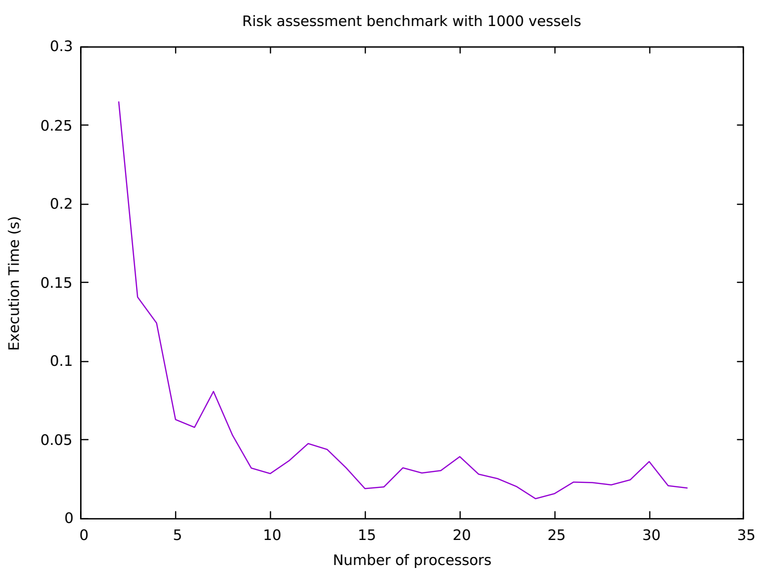

Figure 3: Risk Assessment for 1000 vessels

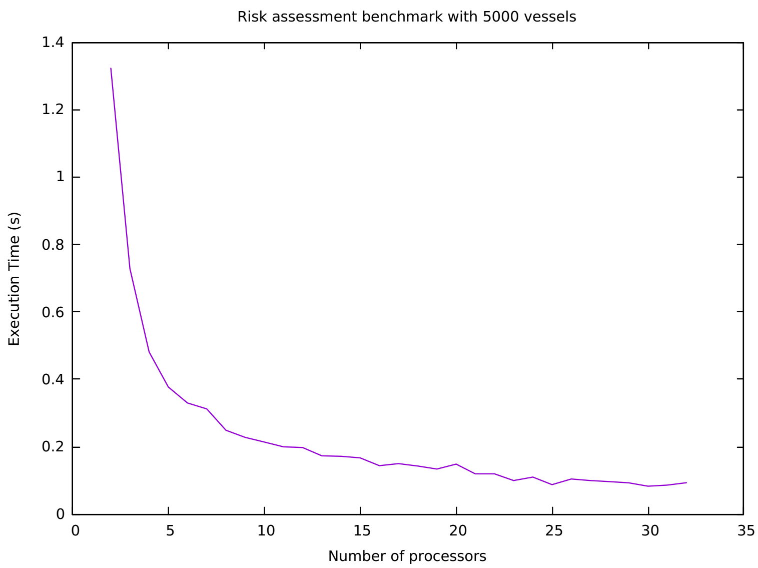

Figure 4: Risk Assessment for 5000 vessels

5 Conclusion

The Risk Management Framework (RMF) is a novel approach for monitoring and managing risk in distributed systems. Wireless sensors report data to a central processing center, which evaluates risk in the environment.

In this study, we have investigated whether or not there is value in augmenting the RMF's "risk extraction" and "risk assessment" modules with the ability to distribute their calculations across multiple CPUs. Furthermore, we have quantified any benefit of doing this.

We have used risk assessment in the maritime domain as a case study for this research. Here the sensors in the network are ships at sea. The vessels broadcast data including their position, speed, and heading which are used by the RMF to calculate risk factors such as the risk of two vessels colliding. For the purpose of this study, we arbitrarily set the maximum acceptable processing time to one second.

Our experimental results show that the processing time of risk evaluation scales linearly with the number of sensors that the RMF is tracking(see Figure 1). Since the RMF is intended to provide results in real time, in any domain there is a number of inputs such that the risk evaluation step does not return results in an acceptable time.

For the maritime domain case study, we found that with one CPU the RMF could handle $\approx$ 3800 vessels before the risk evaluation took more than one second to conclude.

Our experimental results show that parallel computing is a promising approach to decrease processing time for large number of sensors. We have shown that when the input is large, performance scales linearly with the number of CPUs (see Figure 4). When the input is small, performance tends to increase slightly for a small number of CPUs, but degrades as more CPUs are added (see Figure 2, Figure 3).

References

[1] Rami Abielmona. Tackling big data in maritime domain awareness. *Vanguard Magazine*, Aug/Sep:42–43, 2013.

[2] R. Falcon and R. Abielmona. A response-aware risk management framework for search-and-rescue operations. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, June 2012.

[3] R. Falcon, R. Abielmona, S. Billings, A. Plachkov, and H. Abbass. Risk management with hard-soft data fusion in maritime domain awareness. In *the 2014 Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pages 1–8, Dec 2014.

[4] R. Falcon, R. Abielmona, S. Billings, A. Plachkov, and H. Abbass. Risk management with hard-soft data fusion in maritime domain awareness. In *the 2014 Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pages 1–8, Dec 2014.

[5] R. Falcon, A. Nayak, and R. Abielmona. An evolving risk management framework for wireless sensor networks. In *2011 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proceedings*, pages 1–6, Sept 2011.

[6] M. Pothitos, M. Tummala, J. Scrofani, and J. McEachen. Multi-sensor image fusion and target classification for improved maritime domain awareness. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1170–1177, July 2016.

[7] X. Tan, Y. Zhang, X. Cui, and H. Xi. Using hidden markov models to evaluate the real-time risks of network. In *Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008. IEEE International Symposium on*, pages 490–493, Dec 2008.

[8] Leonardo Vanneschi, Mauro Castelli, Ernesto Costa, Alessandro Re, Henrique Vaz, Victor Lobo, and Paulo Urbano. *Improving Maritime Awareness with Semantic Genetic Programming and Linear Scaling: Prediction of Vessels Position Based on AIS Data*, pages 732–744. Springer International Publishing, Cham, 2015.

[9] Yuping Wang, Yiu-ming Cheung, and Hailin Liu, editors. *Computational Intelligence and Security, International Conference, CIS 2006, Guangzhou, China, November 3-6, 2006, Revised Selected Papers*, volume 4456 of *Lecture Notes in Computer Science*. Springer, 2007.

[10] Jinfen Zhang, ngelo P Teixeira, C. Guedes Soares, Xinping Yan, and Kezhong Liu. Maritime transportation risk assessment of tianjin port with bayesian belief networks. *Risk Analysis*, 36(6):1171–1187, 2016.