Reliability Model for Distributed Remote Sensing Application

Tiranee Achalakul and Naruemon Wattanapongsakorn
Department of Computer Engineering, Faculty of Engineering,
King Mongkut's University of Technology Thonburi,
91 Pracha-Uthit Road, Bangkok 10140, Thailand
Tel. +66-2-470-9380, Fax. +66-2-872-5050

Abstract: This paper discusses a software reliability model for the distributed s-PCT algorithm for remote sensing applications. The distributed algorithm is designed based on a Manager-Worker threading concept and goes further to use redundancy to achieve fault tolerance. The paper provides a status report on our progress in developing the reliability concept and applying it to create a model for the distributed s-PCT. In particular, we are interested in the algorithm performance versus reliability.

1. Introduction

Software system reliability is defined as the probability that a system can function flawlessly using specified resources, during a period of time. There are several types of errors that can cause a software system to fail. Discrepancy between expected and actual output is considered one of them. Network and hardware failure can interrupt the software operation and, thus can cause the system to fail. Failure can also come from information attacks, a malicious intention to bring down the system. To equip the software system with a mechanism to provide fault tolerance, threads and processes redundancy are introduced. Applying redundancy will allow a system to gracefully degrade to a point of failure. Redundancy, however, requires additional resources, which in turn implies, longer operation time and higher system cost.

In previous researches, several reliability models have been developed aiming at answering the question of how to introduce redundancy for reliability improvement while maintaining an acceptable performance and cost [8, 11, 15]. In the study by Berman and Ashrafi [4], four types of reliability optimization models are described, namely, one-function system without redundancy, one-function system with redundancy, k-function without redundancy, and k-function with redundancy. Distinctive models can be derived from each type of systems. In this paper, we based our reliability model on the k-function system with redundancy. We propose a reliability model for the distributed spectral screening principal component transform algorithm. This algorithm has been employed in a variety of remote sensing applications including hyper-spectral data compression, information extraction and fusion [12] and change detection [5, 14, 10]. We incorporate the concept of reliability to the algorithm to provide intrusion tolerance to system failure or information attacks using the notion of redundancy. The next section briefly describes the distributed algorithm. Then we present formulations for reliability model. The final section offers concluding remarks.

2. Distributed Spectral-Screening PCT Algorithm

Our application of interest is in the field of remote sensing. The goal is to produce a single color-composite image that represents all the useful information from a set of images from different sensors/wavelengths (multi-spectral image). The data set explored was a 210-channel multi-spectral image, where each image frame corresponds to a foliated scene taken from an altitude of 2000 meters at wavelengths between 400 nm and 2.5 microns. The scene contains a camouflage vehicle. We chose to experiment with s-PCT algorithm [1] because of its capability in summarizing and de-correlating images. The end result is an image that shows significantly improved contrast levels. The forested areas show significantly improved detail and the camouflaged vehicle in the lower left corner is significantly enhanced against its background. Post-processing steps can subsequently be applied to detect edges in the image and use structural information to detect and classify the vehicles. Figure 1 shows frames picked from an input image set and an output image.

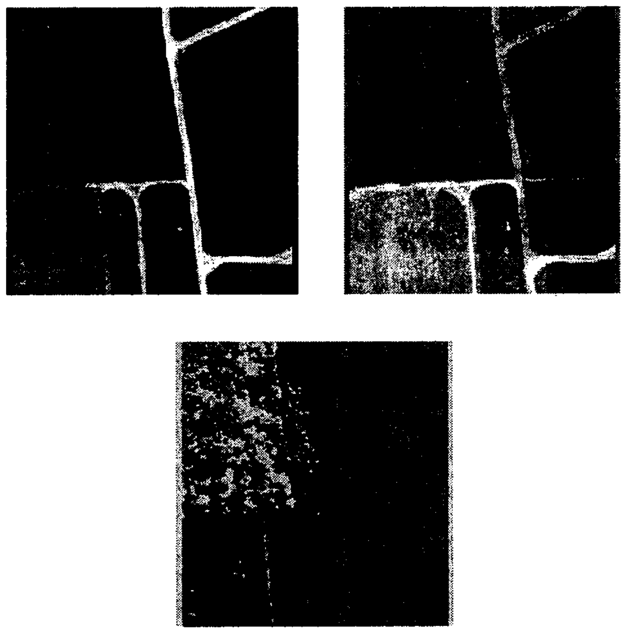

Figure 1. 400, 1998 nm and output images

Our distributed version of the algorithm uses the standard manager/worker decomposition technique [6]. There are three types of threads: The sensor thread partitions the problem and distributes the sub-problems to worker threads. The workers solve each allocated sub-problem, send a result to the manager, and wait for the next sub-problem. A manager thread coordinates the actions of the workers, gathers partial results from them, assembles the final color composite image, and provides access to a displayed hardware. To reduce communication overhead, a worker overlaps the request for its next sub-problem with the calculation associated with the current sub-problem. Using this approach the s-PCT algorithm is divided into 8 steps as follows:

1. ***Spectral classification***: The sensor divides an original image cube into *P* parts, where *P* is the number of workers in the system. Each part, which consists of a set of pixel vectors, is sent to a worker. Each worker operates concurrently to form a unique spectral set by calculating an arccosine of dotproduct of all pixel vector pairs.
2. ***Merge unique sets***: The P unique sets are sent back to the manager and combined. Upon completion there will be one unique set left with K pixel vectors.
3. ***Mean vector***: Each component of the mean vector, m, is the average of the pixel values of each spectral band of the unique set. The n-band image produces a mean vector of n elements.
4. ***Covariance sum***: All the pixel vectors in a unique set are divided into P parts, and sent to P workers. Each worker then computes the covariance component and form a covariance sum.
5. ***Covariance matrix***: The covariance matrix is the average of all the matrices calculated in step 4, and is calculated sequentially by the manager since its complexity is related only to the number of workers rather than the image size.
6. ***Trasformation matrix***: The eigenvectors of the covariance matrix are calculated and sorted according to their corresponding eigenvalues, which provide a measure of their variances. As a result, the high spectral content is forced into the front components. Since the degree of data dependency of the caculation is high, but its complexity is related to the number of spectral bands rather than the image size, this step is also done sequentially.
7. ***Transformation of the data***: Each pixel vector in the original multi-spectral image can be transformed independently. Therefore, all workers transform their portions of the data concurrently.
8. ***Color mapping***: Each worker performs the human-centered color mapping [1] using the first three resulting components of step 7 to generate a portion of the final color image.

3. Reliability Model

In our distributed s-PCT algorithm, sensor, manager, and workers threads are dependent on one another. In other words, all threads must function for the system to function. This is called serial reliability configuration [8]. In this *configuration, the system reliability* R_t can be calculated as follows:

$$R_t = R_s \times R_m \times \prod_{w=1}^{n} R_w \qquad \text{(eq 1)}$$

where R_s = reliability of sensor thread
R_m = reliability of manager thread
R_w = reliability of each worker thread *w*

For each tread i, reliability with constant failure rate can be written as:

$$R_i = e^{-\lambda_i t} \qquad \text{(eq 2)}$$

where $\lambda_i = 1/MTBF_i$ (Mean-Time-Between-Failure)
t = Time of experiments

In serial configuration, there are many points that can cause failure in the system. In order to increase tolerance against hardware/network/software failure and information attacks, redundancy must be introduced. To achieve this, we consider series-parallel reliability model, where the system contains threads with both serial and parallel relationships. Each type of threads has shadow threads to guard against all type of failures. When a thread fails, its shadow can resume the work. Figure 2 shows a system diagram with degree of redundancy equal to two.

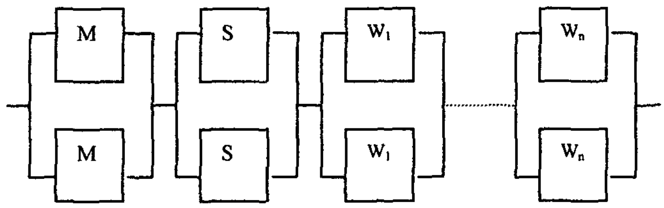

Figure 2. A system comprised of threads in both series and parallel relationship

To calculate reliability with shadow threads, equation 1 can be modified as follows:

$$R_t = (1-(1-R_{s1})(1-R_{s2})) \times (1-(1-R_{m1})(1-R_{m2})) \times \prod_{w=1}^{n} (1-(1-R_{w1})(1-R_{w2})) \qquad \text{(eq 3)}$$

With redundancy of degree *x*, it is expected that the performance would decrease by at least a factor of *x* since the replicated thread required both memory and processor resources. Moreover, the overhead of communication associated with the more complex communication protocols required to achieve redundancy is expected to bring down the performance even further. In our previous experiments, communication cost is approximately 10% of the total time used [2]. The tradeoff between performance and reliability is application dependent. Some applications require a high degree of reliability while others have performance constraints. In this paper, we present a reliability model, which provides a preliminary evaluation of performance versus reliability and redundancy involving in thread replication.

Using techniques developed by Foster et.al [9], a performance prediction model of the s-PCT algorithm can be derived as follows [3]:

$$T_t = \frac{k}{p}(C_1 m^2 sn + C_2 sn + C_3 n^2 s + C_4 n^2 m^2 + C_5 m^2) + C_6(p-1)sn + C_7 n^2 p + C_8 n^3 + C_9 T_w \frac{km^2 n}{p} + T_o \quad \text{(eq 4)}$$

Where T_t is the total execution time on an $n \times m \times m$ image size and s is the number of unique spectra in the spectral screening process. T_w represents the network bandwidth, p is the number of processors used, and k represents granularity level. Using stepwise linear regression methods to eliminate terms that are insignificant to the total time, equation 4 can be re-written as:

$$T_t = C_1 \frac{km^2 sn}{p} + C_2 \frac{ksn}{p} + C_3 \frac{km^2 n^2}{p} + C_4(p-1)sn + C_5 n^3 + T_0 \quad \text{(eq 5)}$$

The constant T_0 and coefficients C_1-C_5 are calibrated using a least-square fitting method and the values are listed below:

$T_0 = 8.462$, $C_1 = 7.397e\text{-}009$, $C_2 = -1.61e\text{-}004$,
$C_3 = 4.438e\text{-}008$, $C_4 = 1.381e\text{-}005$, $C_5 = 6.754e\text{-}006$

As stated in our previous work [2] that the communication overhead for replication is 10% of the total time used, performance cost, C_p ,can be written as follows:

$$C_p = 1.1 T_t Deg_r \quad \text{(eq 6)}$$

Where Deg_r is the degree of redundancy.

The performance cost function is speculated to have an exponential behavior and will act as a penalty of reliability increasing [13]. To obtain the actual relationship between performance cost, C_p, and system reliability, R_t, in our system, an empirical study is performed.

From a previous research by Dugan [7], a typical failure rate of a standard processor is 10^{-5} and the average software failure rate is 10^{-3}. For an experimental period of 1500 hours, we can calculate the reliability using equation 2: the software reliability is 0.8607 and the hardware reliability is 0.9851. Thus, the reliability of each node in the system is 0.8607x0.9851=0.8479. In the remote sensing application described in section 2, a pre-collected data set is used as an input. The sensor reliability, thus, refers to the reliability of a typical processor. In the case of real-time applications, the sensor reliability will depend on the actual sensor used.

We perform an experiment on an image size of $1280 \times 320 \times 210$, T_w = 0.008 microsecond (100BaseT Network), and s = 200. Two, eight, sixteen and twenty-four processors are used with degree of redundancy equal to 1, 2 and 3. Using equations 3 and 6, the reliability model can be formulated:

$$C_p = f(R_t) = ae^{b.R_t} \quad \text{(eq 7)}$$

Applying nonlinear regression technique to equation 7, we can calculate constants a and b, obtaining the values of 128.38 and 1.906 repectively. The sample coefficient of determination (r^2) is 0.68 and the standard error of the estimation is 1.64 seconds. The r^2 of 0.68 is relatively low, which implies that the model may need some improvement. However, in the time range of our problem size, the standard error of 1.64 is acceptable. The predictive data is shown in Figure 3.

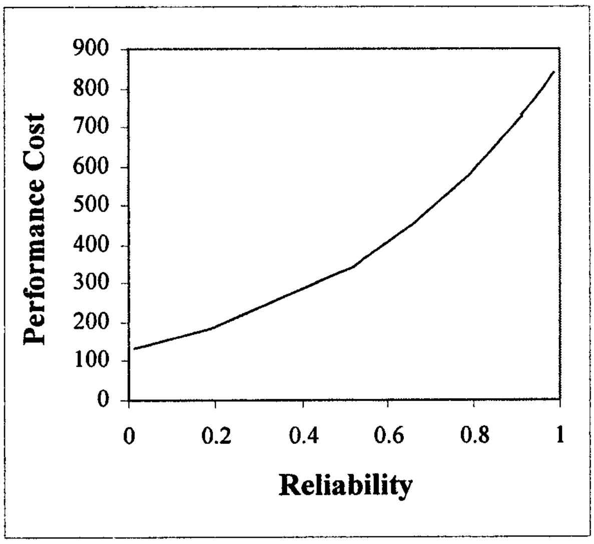

Figure 3. Cost Function Versus Reliability

The low performance cost occurs when a large number of processors are used in the algorithm with none or very little redundancy. However, the reliability will drop significantly with more processors added. Using our predictive model we can find the optimum point of the cost function. In other words, we can predict the performance cost given the minimum reliability required for the system. For example, with the minimum reliability of 0.8, our problem cannot be solved in less than 500 seconds of execution time. The concept of the reliability model developed in this paper can be utilized in designing a system that can realize a performance-reliability objective.

4. Conclusion

This paper has described a fault-tolerant version of our distributed spectral-screening PCT algorithm and its associated analytical model for reliability prediction. The algorithm has been applied to a typical remote sensing data set. The reliability model was calibrated against an experimental data. Given a problem size and a reliability constraint, the model can be used to estimate the achievable performance. Our model is as yet in an early stage of development and is not optimized. Considerable research remains to examine the concept further.

References

[1] (a) Achalakul T. and Taylor S., A Concurrent Spectral-Screening PCT Algorithm for Remote Sensing Applications, *Journal of Information Fusion*, Vol. 1:2, (2000), pp. 89-97.

[2] (b) Achalakul T., Lee J., and Taylor S., "Resilient Image Fusion", *Proceedings of the 2000 International Conference on Parallel Processing Workshops*, Toronto, Canada, pp. 291-296, August 2000.

[3] Achalakul T., and S. Taylor, Real-Time Multi-spectral Image Fusion, *Journal of Concurrency and Computation: Practice and Experience*, Vol. 13:12 (2001), pp.1063-1081.

[4] Berman O., Ashrafi N., Optimization Models for Reliability of Modular Software Systems, *IEEE Transaction on Software Engineering*, Vol. 19:11, (1993), pp. 1119-1123.

[5] Byrne G. F., P. F. Crapper, and K. K. Mayo, Monitoring Land-Cover Change by Principal Component Analysis of Multitemporal Landsat Data, *Remote Sensing of Environment*, Vol. 10, (1980), pp. 175-184.

[6] Chandy K. M., and S. Taylor, *An Introduction to Parallel Programming, Jones and Bartlett Publishers*, Boston, MA, 1992.

[7] Dugan, J. B., "Reliability Analysis of a Hardware and Software Fault Tolerant Parallel Processing", IEEE 13th Symposium on Reliability Distributed Systems, 1994, pp. 74-83.

[8] Ebeling, C. E., *Reliability and Maintainability Engineering: An Introduction,* New York, NY: McGraw-hill, 1997.

[9] Foster I., W. Gropp, and R. Stevens. The Parallel Scalability of the Spectral Transform Method, *Monthly Weather Review*, 1992.

[10] Fung T., and E. LeDrew, Application of Principal Components Analysis to Change Detection, *Photogrammetric Engineering and Remote Sensing*, Vol. 53:12, (1987), pp. 1649-1658.

[11] Helander, M. E., Zhao M., and Ohisson N., Planning models for software reliability and cost, *IEEE Transaction on Software Engineerin,* Vol. 24:6, (1998), pp. 420-434.

[12] Lee J., and K. Hoppel, Principal Components Transformation of Multifrequency Polarimetic SAR Imagery, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 30, (1992), pp. 686-696.

[13] Mettas A., "Reliability Allocation and Optimization for Complex System", Proceedings of Annual Reliability and Maintainability Symposium, Los Angeles, CA, pp. 216-221, January 2000.

[14] Richards J. A., Thematic Mapping from Multitemporal Image Data Using the Principal Components Transformation, *Remote Sensing of Environment*, Vol. 16, (1984), pp. 35-46

[15] Wattanapongsakorn, N. and Levitan, S., "Reliability Optimization Models for Fault-Tolerant Distributed Systems", Proceedings of Annual Reliability and Maintainability Symposium, Philadelphia, PA, pp. 193-199, January 2001.