Summary - Set Reconciliation with Nearly Optimal Communication Complexity

Bowen Song[1]
May 28, 2018

***Abstract*— This summary is a part of an independent study for set and string reconciliation in a distributed system. The summary is reflecting a study of set reconciliation techniques based on polynomial interpolation method[1]. A further comparison to other techniques will be available. This report presents reviews and discussion over set reconciliation protocols presented by the paper.**

I. Introduction

This paper provides an approach to a set reconciliation problem defined as reconciling two sets of similar data under minimum communication and computation complexity. In addition, the paper specifies a set reconciliation problem as two physically separated similar sets under limited network connectivity. Such setup describes system like a common distributed database or file system where synchronization process is not on the priority network[2].

II. Overview

The paper uses Polynomial Interpolation as the core of its proposed protocols to set reconciliation problems. The protocols use the anti-entropy approach, fully reconciles set pairs instead of restricting to recent information. The proposed protocol covers set reconciliation with known bound on sets differences as a deterministic method. Without a known bound the protocol extends to a probabilistic method under several assumptions. The goal is to improve the set reconciliation performance in terms of communication complexity. There exist other approaches to set reconciliation problems that consider error-correcting codes and Bloom filters.

The paper has made a connection specifically with error-correcting codes and an analogy to Reed-Solomon codes based on the protocol. The paper points out a high computation complexity generated from standard encoding and decoding algorithms.

III. Polynomial Interpolation

The core part of the protocol[2] is to represent sets by their characteristic polynomials and manipulate based on their values. The common knowledge between two hosts for reconciliation with a known set difference upper bound includes finite field, a sum of set difference upper bound $\bar{m}$, and $\bar{m}$ amount of evaluation points E. The protocol starts with two hosts, A and B, evaluating characteristic polynomials based on all evaluation points. The results are transmitted over to perform finite field division and recover coefficient of reduced rational function. At last factor the rational function to obtain the set differences. All calculations are within the common finite field.

The protocol can also work without a prior knowledge of set difference upper bound by a combination of guessing and verifying. The goal is to guess a large enough upper bound which is a close fit to the actual difference in order to minimize both communication and computation complexities. The paper is suggesting two ways of testing a correct upper bound probabilistically. The first method is to finish entire process of the protocol and compare hashes of the sets to determine if the resulted sets are equal. This method is obviously not the most efficient but very easy for implementation and adequate for small sets. The second method is to test the equality of rational functions after recovering its coefficients.

As two probabilistic methods, the confidence of their verification results relies on hash functions and number of evaluation points as a significant fraction of the finite field given the two sets are sparsed respectively.

IV. Performance Metric

Complexity	Communication	Computation
Known upper bound on sets differences	$O(\bar{m}b)$	$O(\bar{m}^3)$
Without knowing upper bound on sets differences (Minimizing the number of transmitted bits)	$O(b(m+k))$	$O(m^4)$
Without knowing upper bound on sets differences (Minimizing round complexity)	$O(b(m+k))$	$O((m+k)^3)$

k = extra evaluations based on a preset probability of failure
b = Length of bitstrings
m = sum of set differences

[1]B. Song is with Department of Electrical and Computer Engineering, Boston University, Boston MA, sbowen@bu.edu

[2]In this summary, we are focusing on the second protocol from protocol 1 and 2 in the paper. The protocol 1 is very restricted as it only works in the situation where set A and B are different by one element. This is handled by a simple parity sum check.

V. Conclusion

The Communication complexity of this set reconciliation protocol is close to the size of the symmetric difference of the two sets. With a prior knowledge of sets symmetric difference, it can be fit into a broadcasting situation. The protocol can also work without the prior knowledge. The communication complexity of the protocol is proved to be within a small constant to the best achievable.

The computation complexity is fairly high compared to existing work that is less efficient on communication. There is also an assumption for the sparsity of the sets.

The proposed polynomial interpolation based protocol achieves a close to optimal communication complexity for set reconciliation. The protocol can broadcast with known set differences and can handle situations where bound for sets difference is not available. Despite the fairly large computation complexity, the protocol can fit into a modern distributive system where limited network communication is more important than computation limitation.

References

[1] Y. Minsky, A. Trachtenberg, and R. Zippel, "Set reconciliation with nearly optimal communication complexity," *IEEE Transactions on Information Theory*, vol. 49, no. 9, pp. 2213–2218, Sept 2003.

[2] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined wan," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 3–14.