Independent Study Proposal for Set and String Reconciliation in a Distributed System

Bowen Song[1]
May 10, 2018

***Abstract*— This independent study is interested in conducting research in the field of Computer Engineering to prepare for a thesis on set or string reconciliation for distributed systems.**

I. Purpose of Independent Study

The goal of this independent study is to explore various protocols for set and string reconciliation and how they may be used in a distributed system. The base of this independent study is built upon series of work over set and string reconciliation between two hosts with similar collections and under low communication costs. The available research selected for this study is centered around Characteristic Polynomial Interpolation-based Synchronization (CPIsync) scheme for set reconciliation [1] [2] and its closely related work [3] [4]. The study also includes existing methods for string reconciliation and how to fit synchronization processes in Personal Digital Assistant (PDA) architectures and distributed systems. Existing reconciliation methods not limited to using invertible Bloom filter and lookup tables based on other synchronization methods are included in this study. The study also covers *rsync* which will be included in the scope of benchmark testing comparison for the thesis.

II. Motivation

Synchronization in a distributed system between components is one of the most important part of keeping data consistency between devices. Applications such as content distribution and storage networks, that need to maintain a continuous consistent replicas with in a distributed environment under limited bandwidth, would potentially benefit from this study.

III. Choice of Readings

The independent study consists of investigating researches on set and string reconciliation, synchronization on mobile computing devices and distributed system, and other related synchronizing methods and protocols.

i Set reconciliation (Task 1)
  a) Practical Set Reconciliation [1]
  b) Reconciling Graphs and Sets of Sets [2]
  c) Reconciling Similar Sets [3]
  d) Robust Set Reconciliation [4]
  e) Set Reconciliation with Nearly Optimal Communication Complexity [5]

ii Synchronization on PDA architectures (Task 2)
  a) Efficient PDA Synchronization [6]
  b) Fast PDA Synchronization Using Characteristic Polynomial Interpolation [7]
  c) On the Scalability of Data Synchronization Protocols for PDAs and Mobile Devices [8]

iii String reconciliation (Task 3)
  a) Bandwidth Efficient String Reconciliation using Puzzles [9]
  b) Efficiently Decoding Strings from Their Shingles [10]
  c) Reconciliation Puzzles [11]

iv Synchronization on distributed system (Task 4)
  a) Algorithms for Low-Latency Remote File Synchronization [12]
  b) Efficiently Repairing and Measuring Replica Consistency in Distributed Databases [13]
  c) Improved File Synchronization Techniques for Maintaining Large Replicated Collections over Slow Networks [14]
  d) Synchronization and Deduplication in Coded Distributed Storage Networks Distributed Storage Networks [15]

v Synchronization with invertible Bloom filters (Task 5)
  a) Set Reconciliation and File Synchronization Using Invertible Bloom Lookup Tables [16]
  b) Synchronizing Namespaces with Invertible Bloom Filters [17]

vi Other Approaches (Task 6)
  a) An Efficient PGP Keyserver without Prior Context [18]
  b) Data Verification and Reconciliation with Generalized Error-Control Codes [19]
  c) Efficient Reconciliation and Flow Control for Anti-Entropy Protocols [20]
  d) Efficient Point-to-Multipoint Data Reconciliation [21]

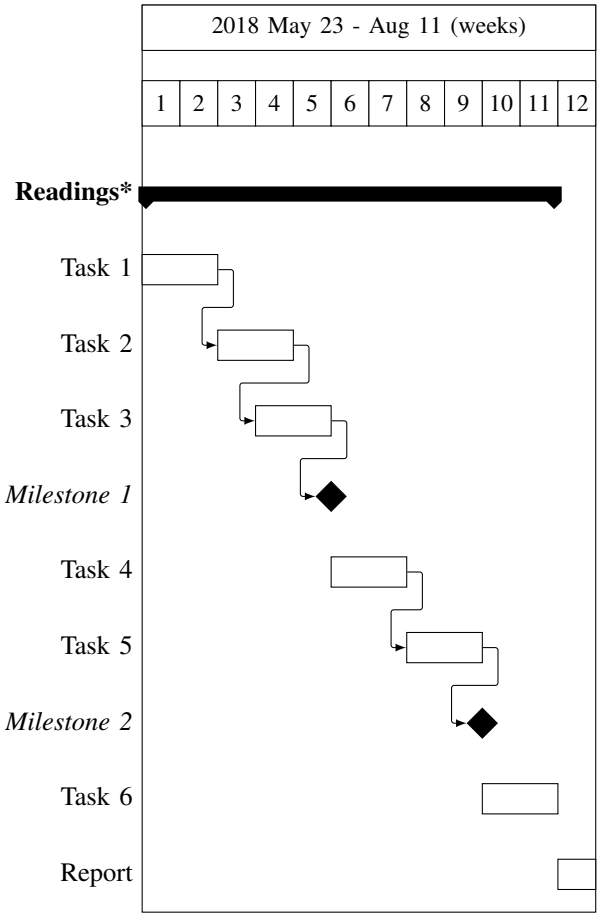
IV. Research Approach and Expectations

This independent study will be paired with experimenting the concepts of the reading materials either by using existing libraries or implementing examples as proof of concepts. The final expectations of this independent study are to have a clear understanding of selected set and string reconciliation

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

protocols and a solution plan for string reconciliation based on CPIsync scheme.

V. Timeline and Millstone



Readings* - Weekly deliverables are to show understanding of learnt subjects by one of presentation, report, or code implementation

Task - Reading tasks that are listed in Section III

Milestone - Deliver report related to learnt materials

References

[1] Y. Minsky and A. Trachtenberg, "Practical set reconciliation," in *40th Annual Allerton Conference on Communication, Control, and Computing*, vol. 248, 2002.

[2] M. Mitzenmacher and T. Morgan, "Reconciling graphs and sets of sets," *arXiv preprint arXiv:1707.05867*, 2017.

[3] R. Gabrys and F. F. Hassanzadeh, "Reconciling similar sets," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Oct 2017, pp. 1139–1144.

[4] D. Chen, C. Konrad, K. Yi, W. Yu, and Q. Zhang, "Robust set reconciliation," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. ACM, 2014, pp. 135–146.

[5] 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.

[6] D. Starobinski, A. Trachtenberg, and S. Agarwal, "Efficient pda synchronization," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 40–51, 2003.

[7] A. Trachtenberg, D. Starobinski, and S. Agarwal, "Fast pda synchronization using characteristic polynomial interpolation," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1510–1519.

[8] S. Agarwal, D. Starobinski, and A. Trachtenberg, "On the scalability of data synchronization protocols for pdas and mobile devices," *IEEE network*, vol. 16, no. 4, pp. 22–28, 2002.

[9] S. Agarwal, V. Chauhan, and A. Trachtenberg, "Bandwidth efficient string reconciliation using puzzles," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 11, pp. 1217–1225, 2006.

[10] A. Kontorovich and A. Trachtenberg, "Efficiently decoding strings from their shingles," *arXiv preprint arXiv:1204.3293*, 2012.

[11] V. Chauhan and A. Trachtenberg, "Reconciliation puzzles [separately hosted strings reconciliation]," in *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, vol. 2. IEEE, 2004, pp. 600–604.

[12] H. Yan, U. Irmak, and T. Suel, "Algorithms for low-latency remote file synchronization," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, April 2008.

[13] J. García-García, C. Ordonez, and P. T. Tosic, "Efficiently repairing and measuring replica consistency in distributed databases," *Distributed and Parallel Databases*, vol. 31, no. 3, pp. 377–411, 2013.

[14] T. Suel, P. Noel, and D. Trendafilov, "Improved file synchronization techniques for maintaining large replicated collections over slow networks," in *Data Engineering, 2004. Proceedings. 20th International Conference on*. IEEE, 2004, pp. 153–164.

[15] S. El Rouayheb, S. Goparaju, H. M. Kiah, and O. Milenkovic, "Synchronization and deduplication in coded distributed storage networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 3056–3069, 2016.

[16] M. Gentili, "Set reconciliation and file synchronization using invertible bloom lookup tables," Ph.D. dissertation, 2015.

[17] W. Fu, H. B. Abraham, and P. Crowley, "Synchronizing namespaces with invertible bloom filters," in *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*. IEEE, 2015, pp. 123–134.

[18] A. Rucker, "An efficient pgp keyserver without prior context," 2017.

[19] M. G. Karpovsky, L. B. Levitin, and A. Trachtenberg, "Data verification and reconciliation with generalized error-control codes," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1788–1793, 2003.

[20] R. Van Renesse, D. Dumitriu, V. Gough, and C. Thomas, "Efficient reconciliation and flow control for anti-entropy protocols," in *proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware*. ACM, 2008, p. 6.

[21] P. Rodriguez and J. Chesterfield, "Efficient point-to-multipoint data reconciliation," Jul. 19 2011, uS Patent 7,984,018.