Adapting MapReduce for Efficient Watermarking of Large Relational Dataset

Sapana Rani, Dileep Kumar Koshley and Raju Halder
Indian Institute of Technology Patna, India
{sapana.pcs13, dileep.pcs15, halder}@iitp.ac.in

TrustCom, 2017

Outline

Outline

Introduction

- **U.S. health care data** - 150 exabytes (10^{18}) in 2011 [Raghupathi and Raghupathi, 2014].

- **Large volume of data** [Russom, 2013]:
  - $> 20\% =$ structured
  - stored in relational databases.
- **Vulnerabilities** [Halder et al., 2010]:
  - copyright infringement
  - data tampering
  - integrity violations
  - illegal redistribution
  - ownership claiming
  - piracy, etc.

Introduction

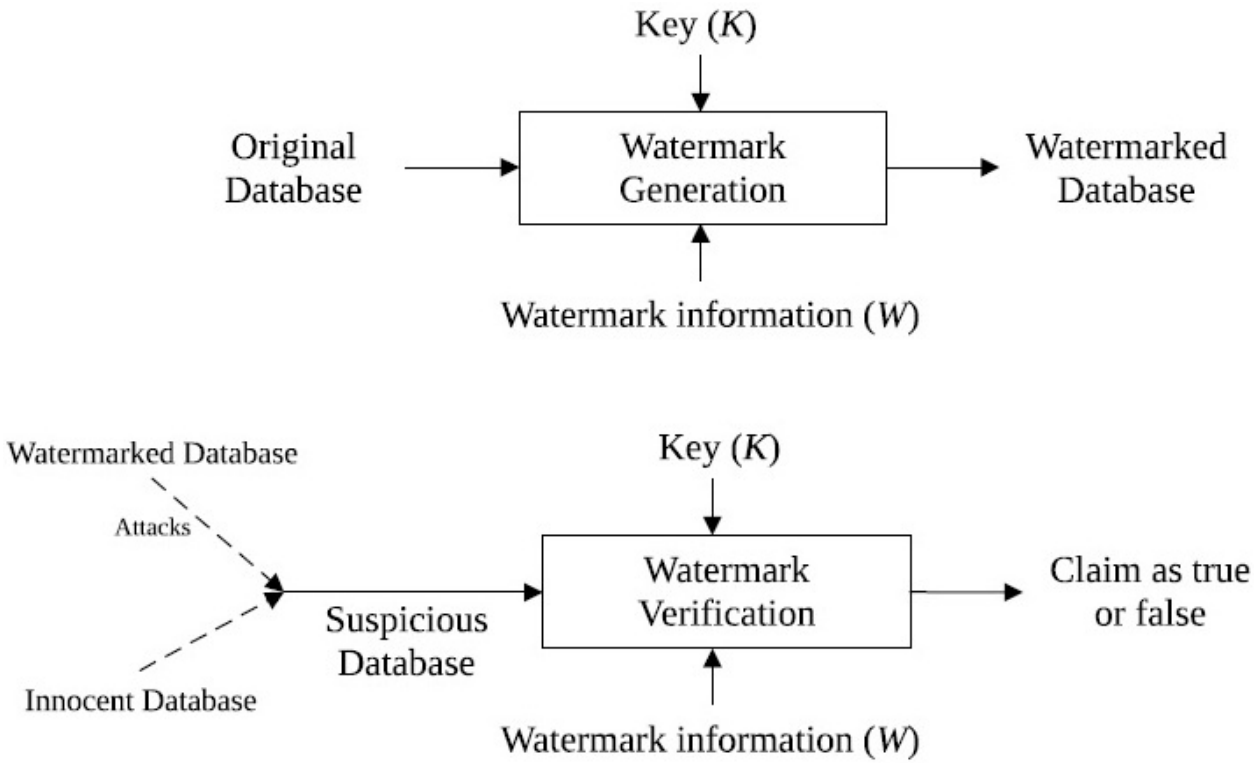
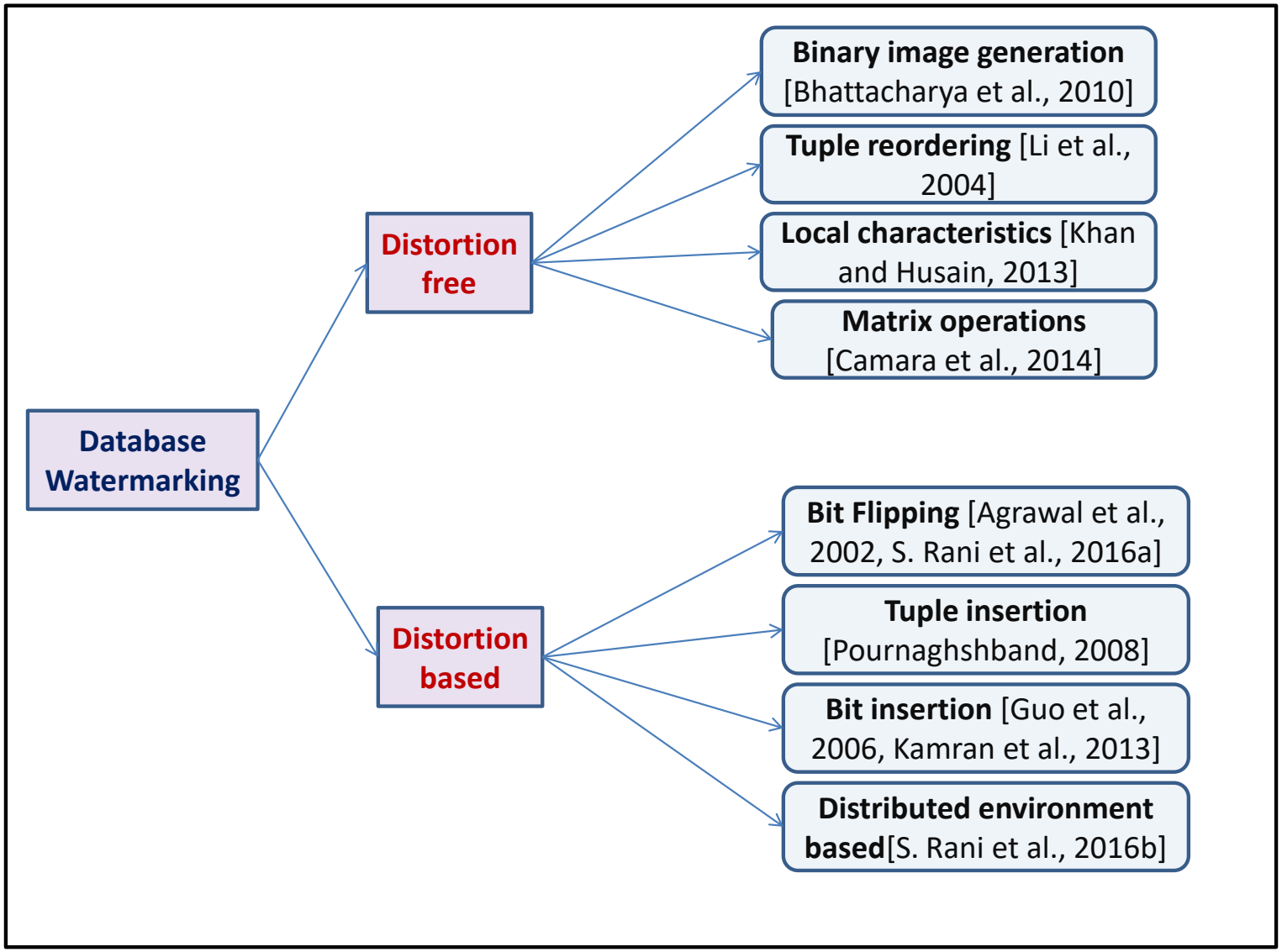


Figure: Database Watermarking

Outline

Related Works



Distortion based schemes

Proposed schemes	Watermark information	Cover type	Verifiability	Intent
AHK algorithm [Agrawal et al., 2002]	Meaningless bit pattern	Numeric	Blind, private	Ownership proof
Fake tuple-based [Pournaghshband, 2008]	Fake info obtained from db content	Database table	Blind, private	Ownership proof
Fake attribute-based [Prasannakumari, 2009]	Database content	Database table	Blind, private	Tamper detection

Distortion free schemes

Proposed schemes	Watermark information	Cover type	Verifiability	Intent
Permutation based [Li et al., 2004] [Bhhattacharya et al., 2009]	A part of group level hash value	Tuples positions	Blind, private	Tamper detection
Binary form relation [Li and Deng, 2006] [Halder and Cortesi, 2010]	Relation in binary form	Nil	Blind, public or private	Tamper detection and ownership proof
Characteristic based [Khan and Husain., 2013]	Frequency distribution of digit count,length and range of data values	Numeric	Blind, private	Tamper detection and nature of tampering
Database Content [Camara et al., 2014]	Determinant and minor of group matrix	Numeric	Blind, private	Tamper detection

Outline

Motivations

- Sequential watermark processing for large scale dataset $\Rightarrow$ Inefficient.
- Periodic rewatermarking $\Rightarrow$ Impractical
- Distributed embedding and detection is important.

Contributions

- Adapted Mapreduce for processing large dataset.
- Presented a case study for transforming sequential to MapReduce based watermarking.
- The experimental results $\Rightarrow$ improvements in performance.

Outline

MapReduce

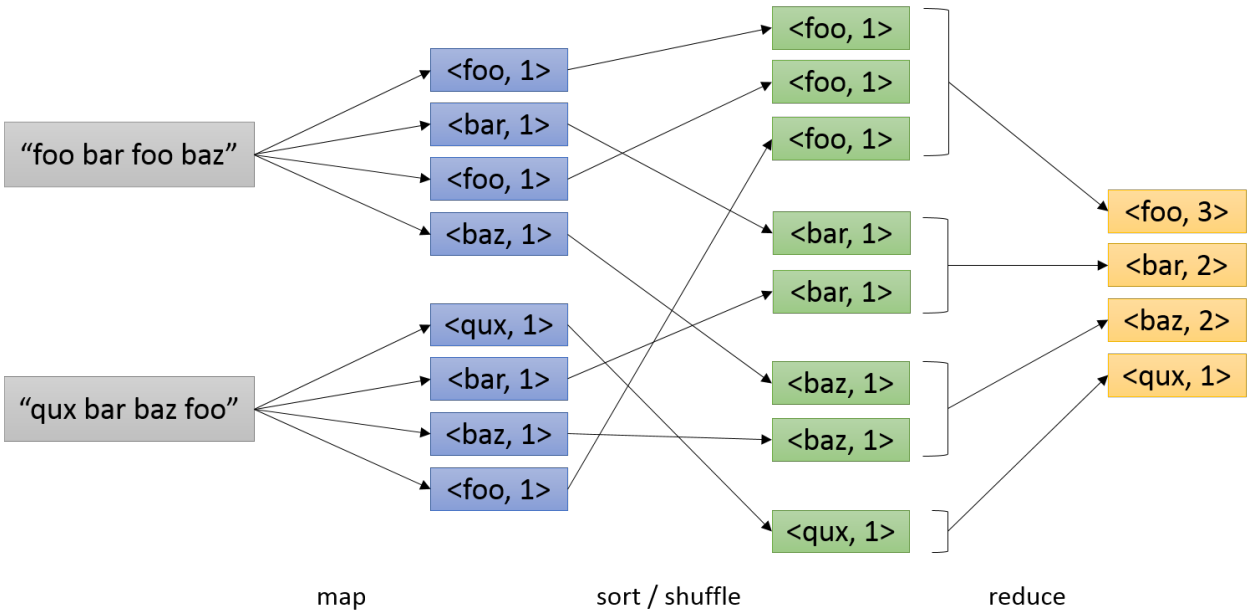


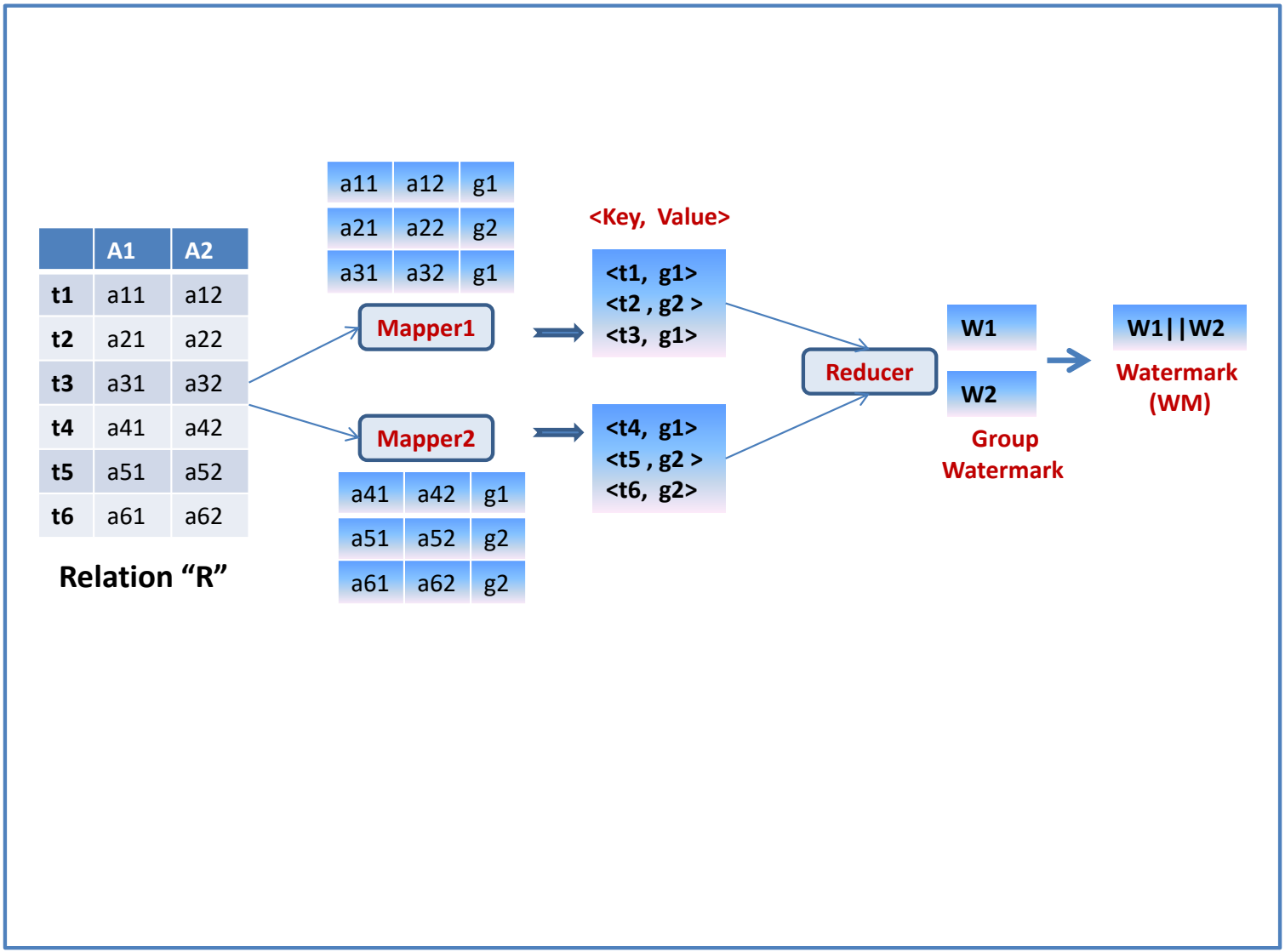
Figure: Word frequency computation [Dean and Ghemawat, 2008]

Outline

Watermark Generation using MapReduce

- We consider distortion-free watermarking
- **Primary phases:**
  1. Partitioning of tuples into groups
  2. Watermark generation from each group
  3. Final watermark - generated by combining the group watermarks.

Watermark Generation using MapReduce



Watermark Generation using MapReduce

Algorithm 1 Watermark Generation: MAPPER

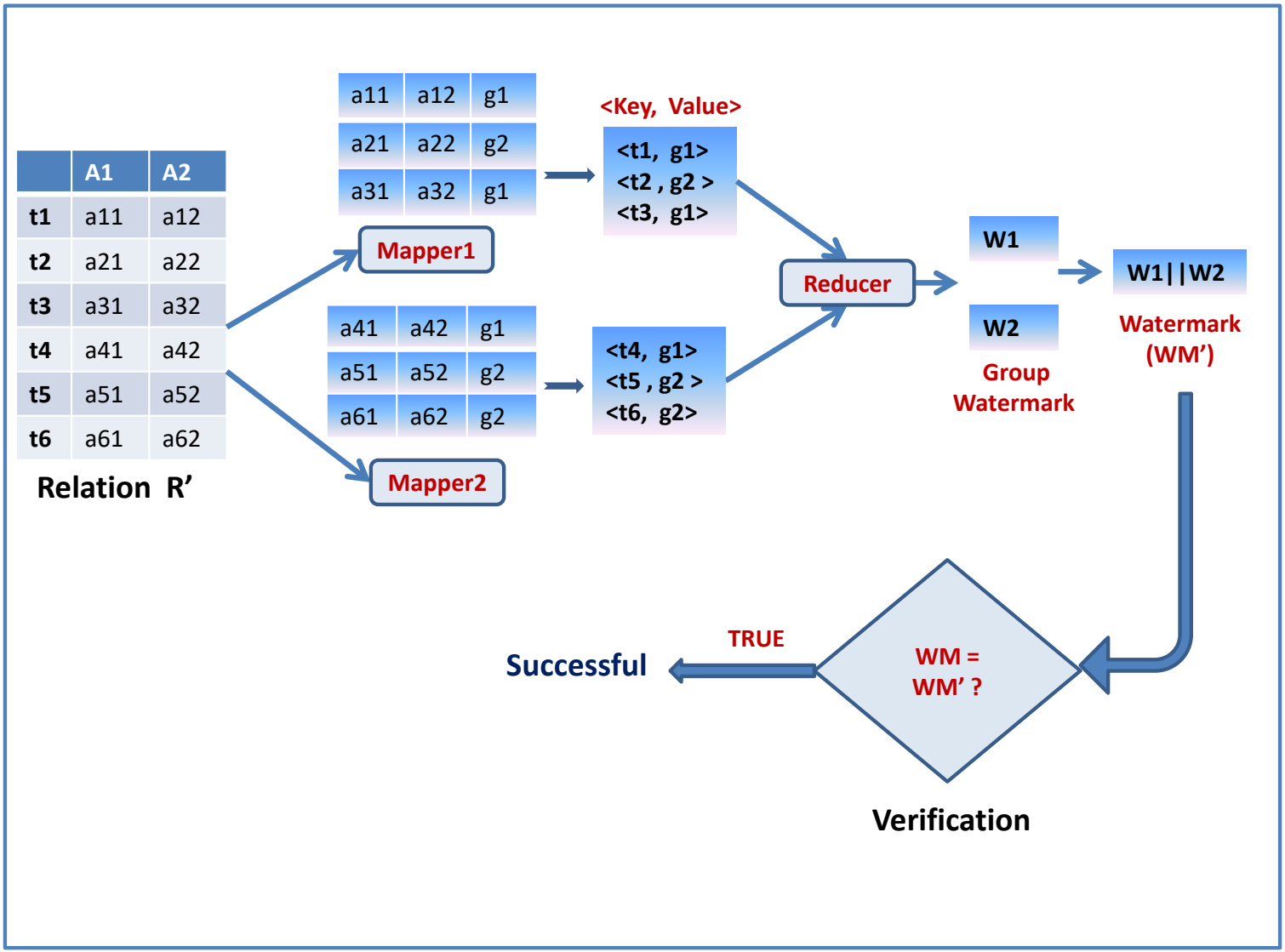
1: **procedure** MAP (R)
2: **for** each tuple $t \in R$ **do**
3: Compute $g_{id} = f(t)$
4: *emit* (g_{id}, t)
5: **end for**
6: **end procedure**

Algorithm 2 Watermark Generation: REDUCER

1: **procedure** REDUCE $(g_{id}, list := [t_1, t_2, ...])$
2: Compute watermark W_g for g_{id} using *list*
3: *emit* (g_{id}, W_g)
4: **end procedure**
5: Compute $W = ||W_g$

Figure: Watermark Generation algorithm using MapReduce

Watermark Detection using MapReduce



Watermark Detection using MapReduce

Algorithm 3 Watermark Detection: MAPPER

1: **procedure** MAP $(R^{'})$
2: **for** each tuple $t \in R^{'}$ **do**
3: Compute $g_{id} = f(t)$
4: *emit* (g_{id}, t)
5: **end for**
6: **end procedure**

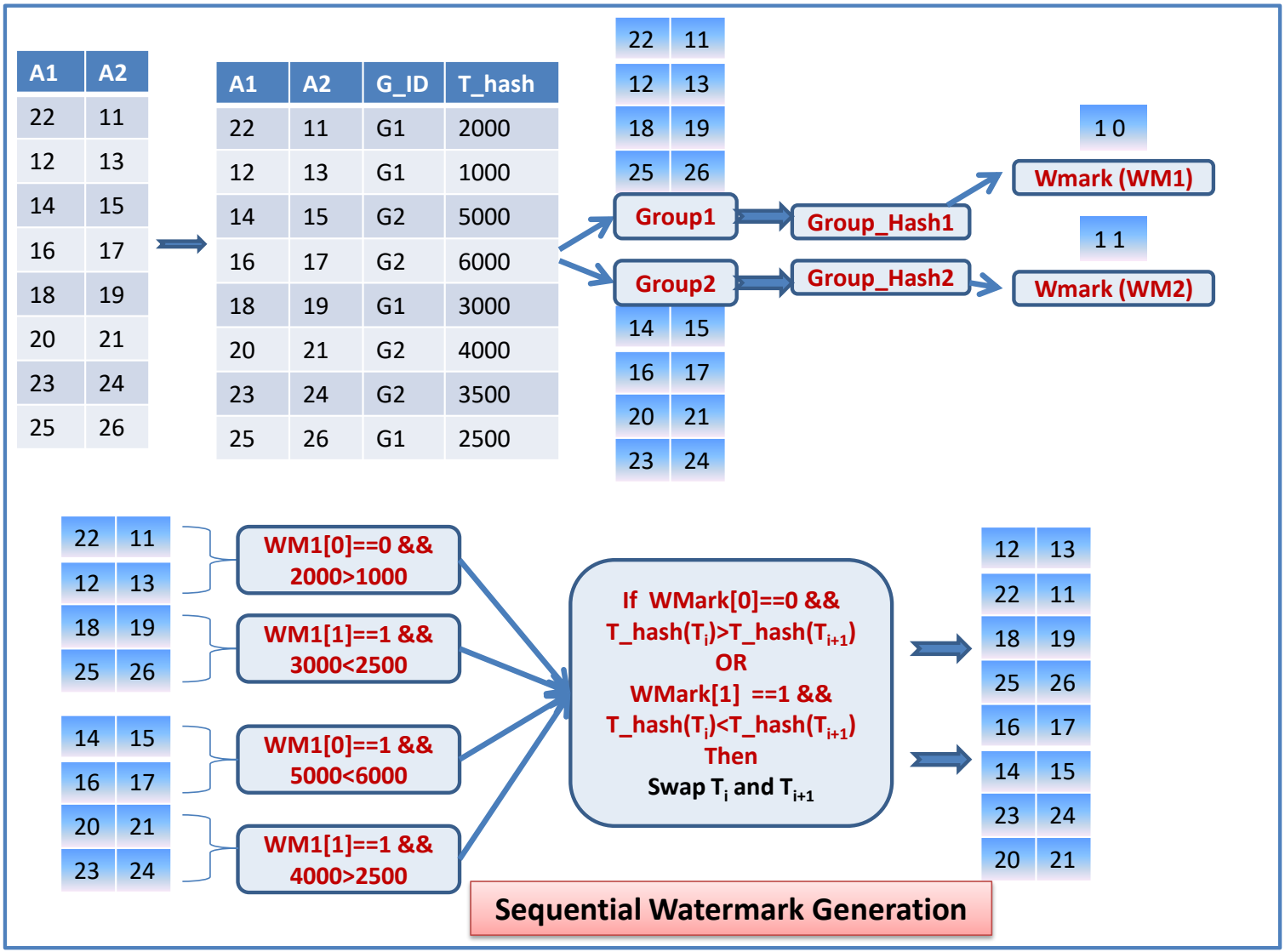
Algorithm 4 Watermark Detection: REDUCER

1: **procedure** REDUCE $(g_{id}, list := [t_1, t_2, ...])$
2: Compute $W_g^{'}$ for g_{id} using *list*
3: *emit* $(g_{id}, W_g^{'})$
4: **end procedure**
5: Compute $W^{'} = ||W_g^{'}$
6: **if** $(W^{'} == W)$ **then** Verification $=$ *TRUE*
7: **else** Verification $=$ *FALSE*
8: **end if**

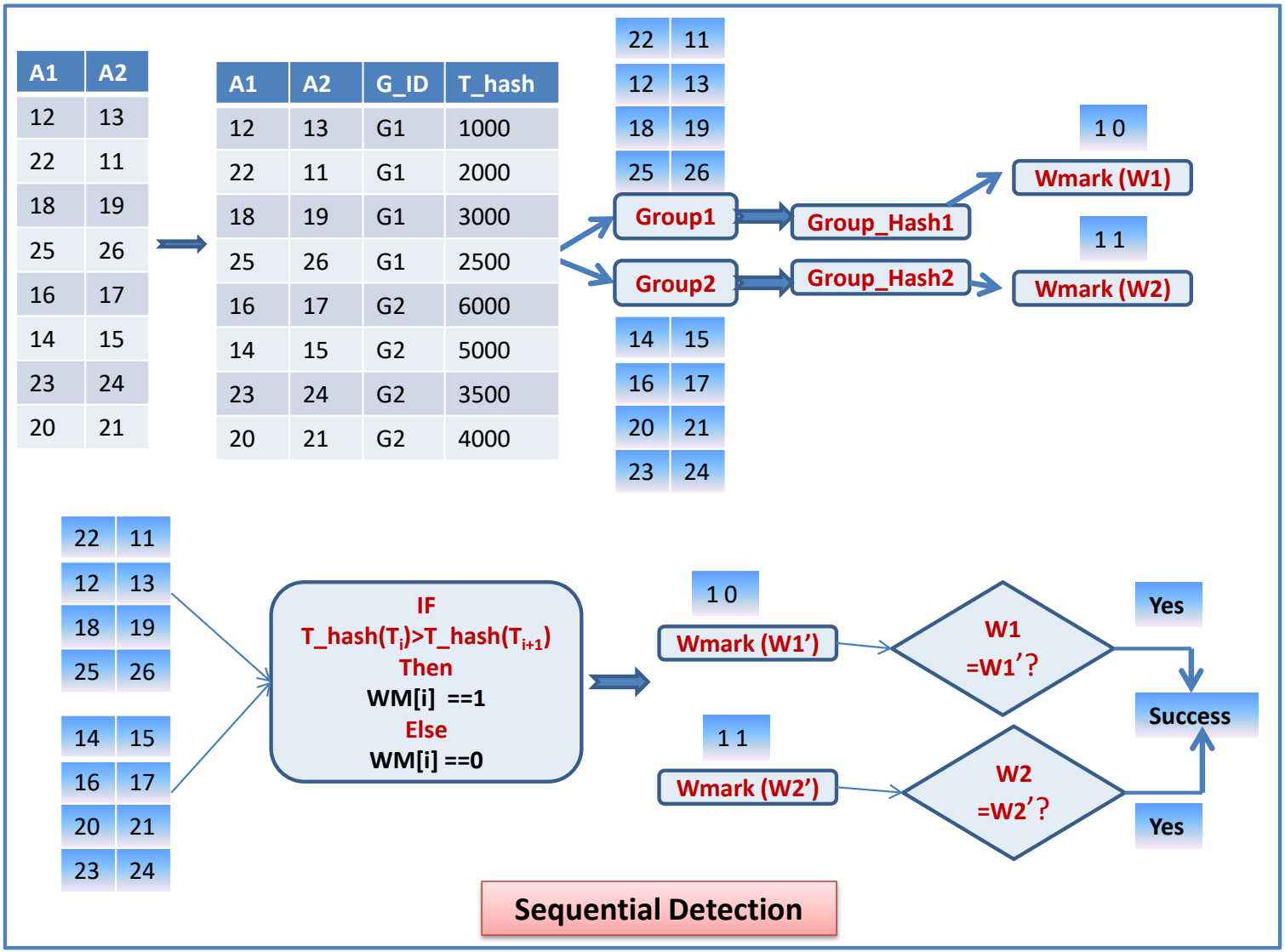
Figure: Watermark Detection algorithm using MapReduce

Outline

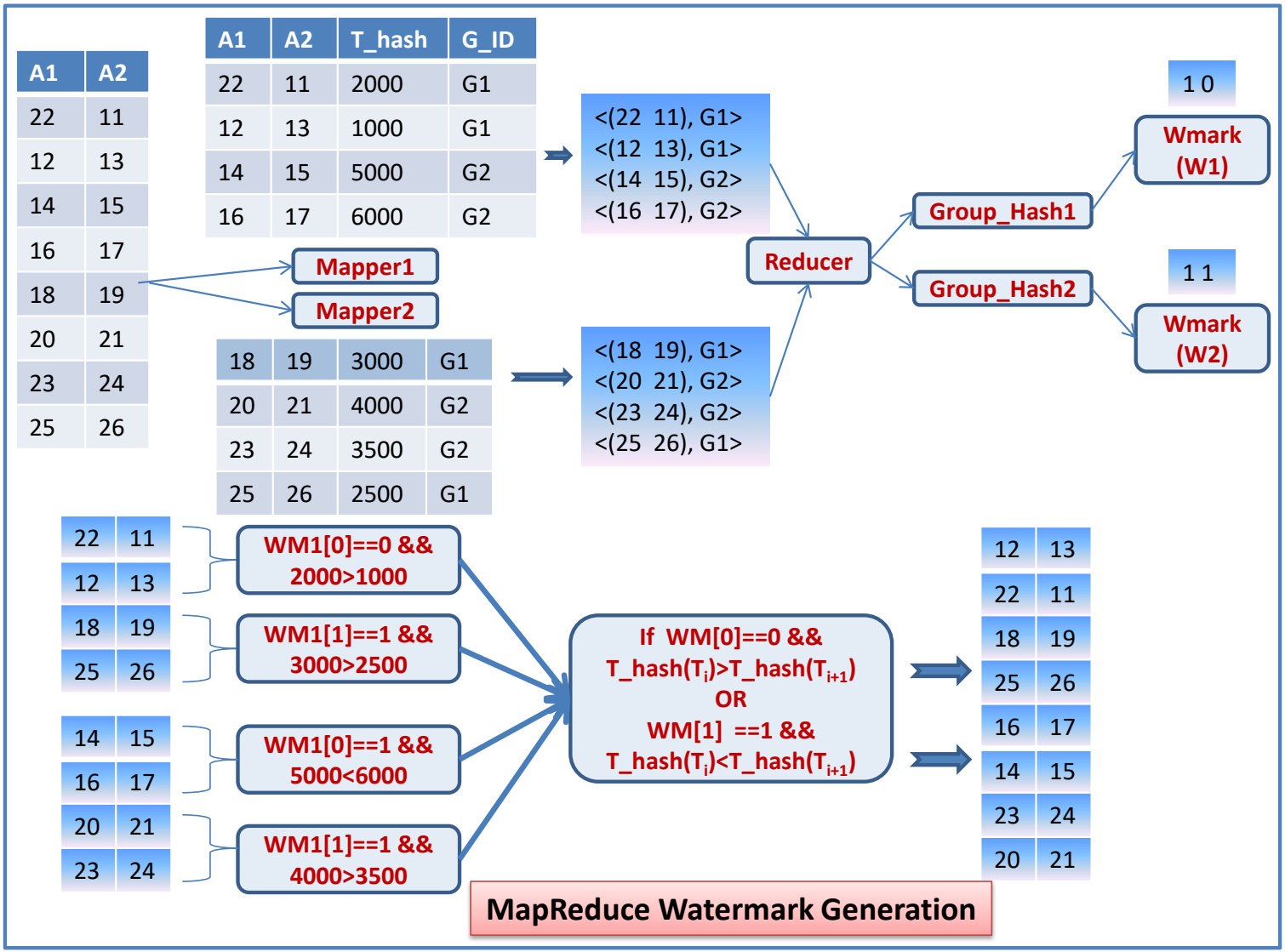
Transforming Sequential to MapReduce [Li et al., 2004]



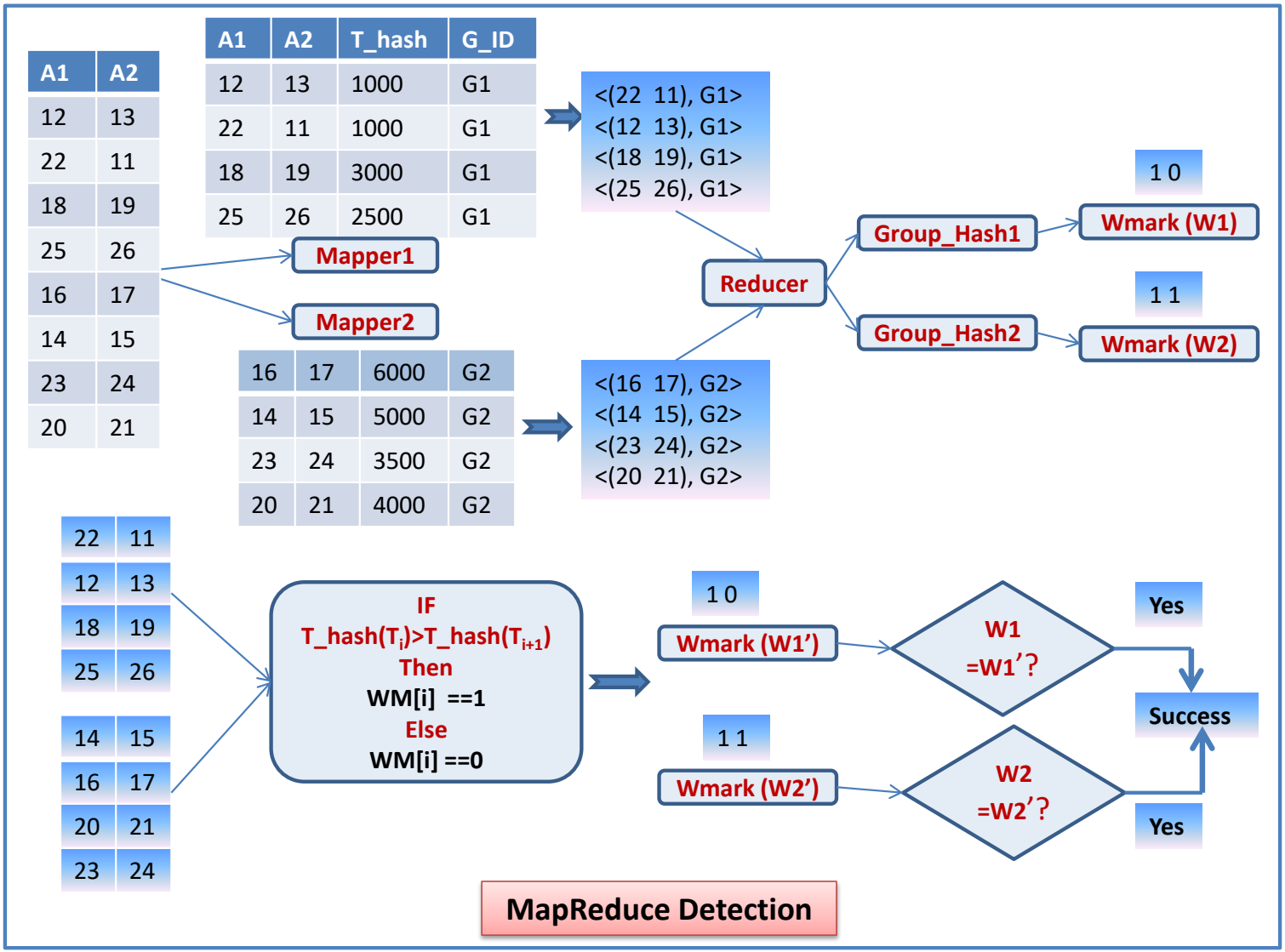
Transforming Sequential to MapReduce [Li et al., 2004]



Transforming Sequential to MapReduce [Li et al., 2004]



Transforming Sequential to MapReduce [Li et al., 2004]



Outline

Experimental Results

- Identified following distortion-free techniques:
  - [Li et al., 2004]
  - [Bhattacharya and Cortesi, 2009]
  - [Bhattacharya and Cortesi, 2010]
  - [Khan and Husain, 2013]
  - [Camara et al., 2014]

- **Notations:**

T_g : Watermark generation time in minute

T_d : Watermark detection time in minute

M : Number of Mappers

G : Number of groups formed from the database tuples

Experimental Results: [Li et al., 2004]:

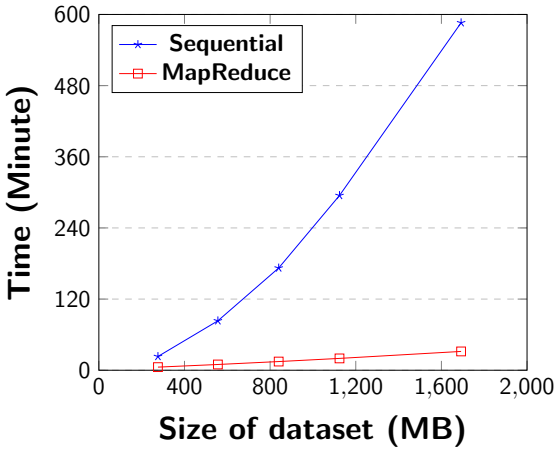
No. of Tuples	Size in MB	Sequential		MapReduce		
		T_g	T_d	M	T_g	T_d
5810120	276	23.19	24.43	2	5.26	5.23
11620240	556	83.36	75.62	5	9.70	9.18
17430360	840	172.44	169.66	7	14.64	13.44
23240480	1124	294.74	291.60	9	19.96	18.28
34860720	1692	586.00	566.83	13	31.72	28.61

(a) Watermarking results for G = 10000

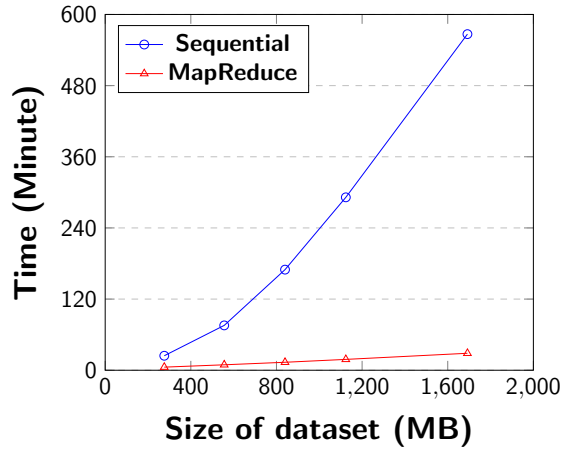
No. of Tuples	Size in MB	Sequential		MapReduce		
		T_g	T_d	M	T_g	T_d
5810120	276	7.16	6.87	2	5.11	4.89
11620240	556	22.99	19.22	5	9.26	8.52
17430360	840	41.75	39.08	7	13.55	12.25
23240480	1124	69.19	66.60	9	18.26	16.23
34860720	1692	124.02	120.25	13	25.64	23.83

(b) Watermarking results for G = 50000

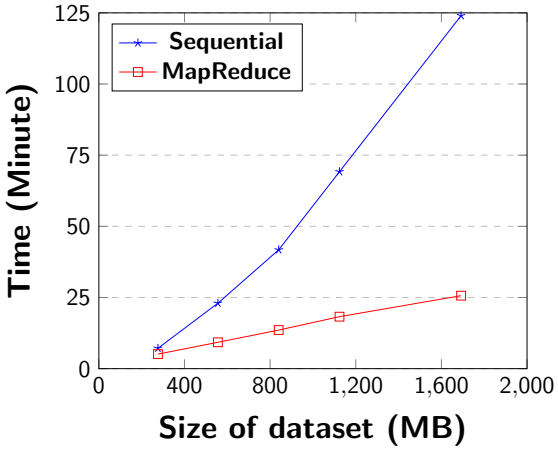
Experimental Results: [Li et al., 2004]:



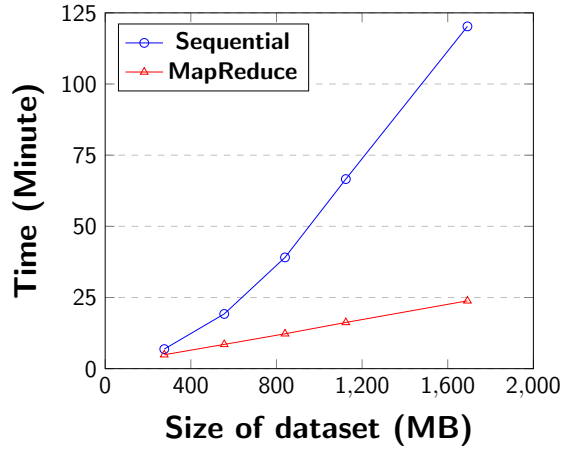
(a) Watermark Generation Time for G=10000



(b) Watermark Detection Time for G=10000

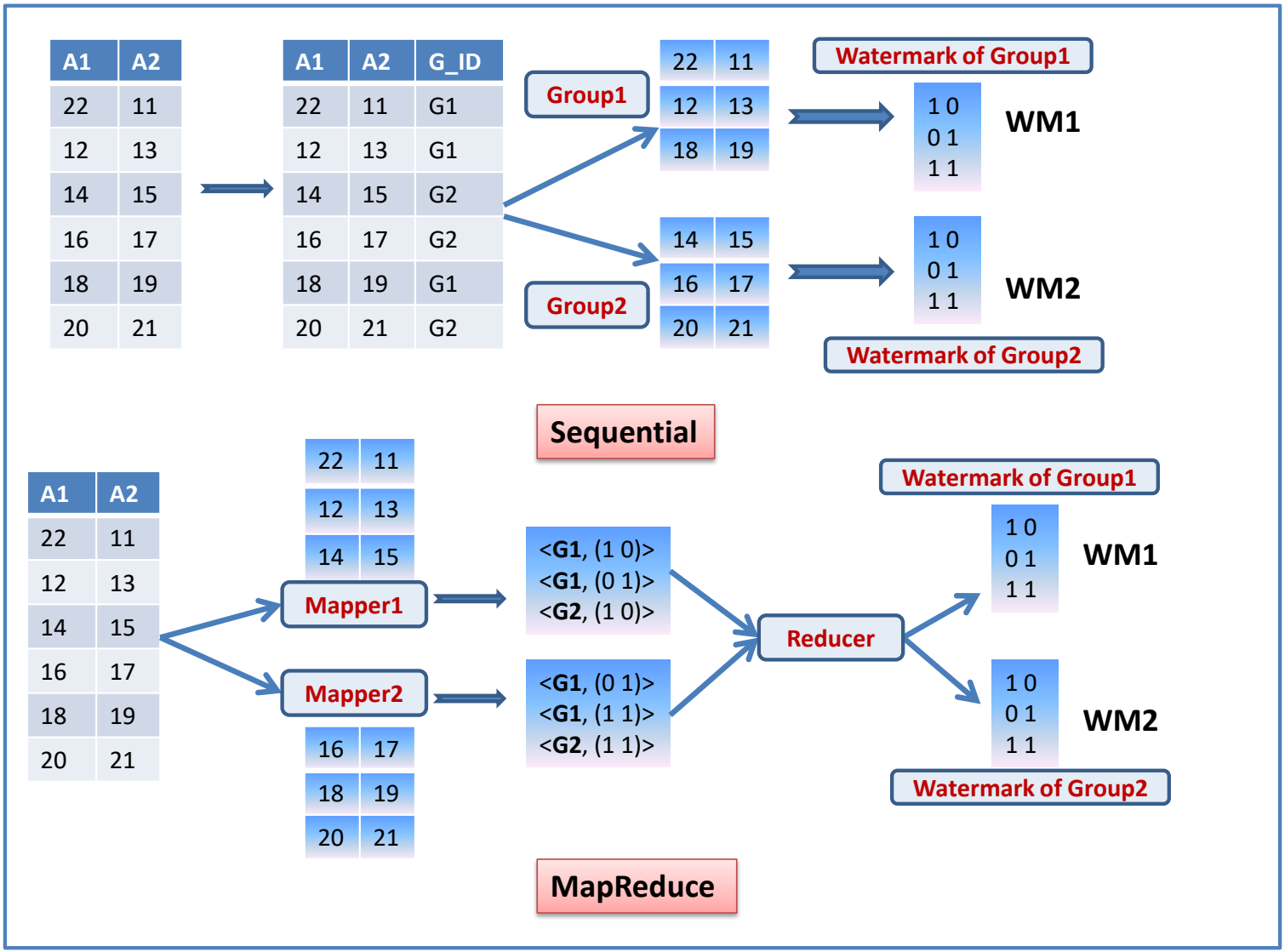


(c) Watermark Generation Time for G=50000



(d) Watermark Detection Time for G=50000

Experimental Results: [Bhattacharya and Cortesi, 2009]

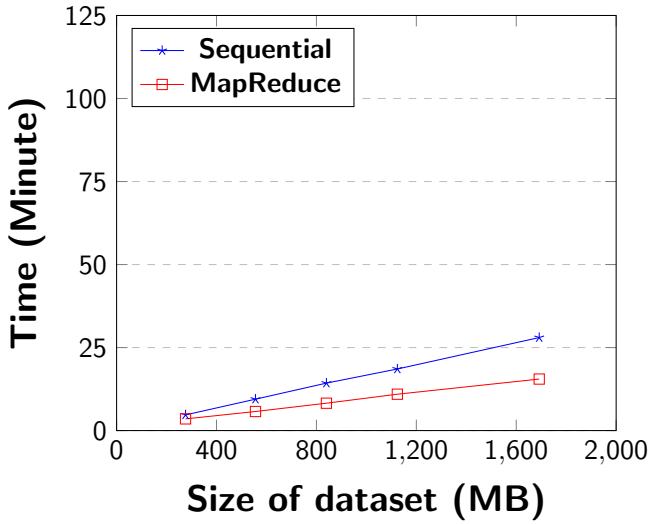


Experimental Results: [Bhattacharya and Cortesi, 2009]

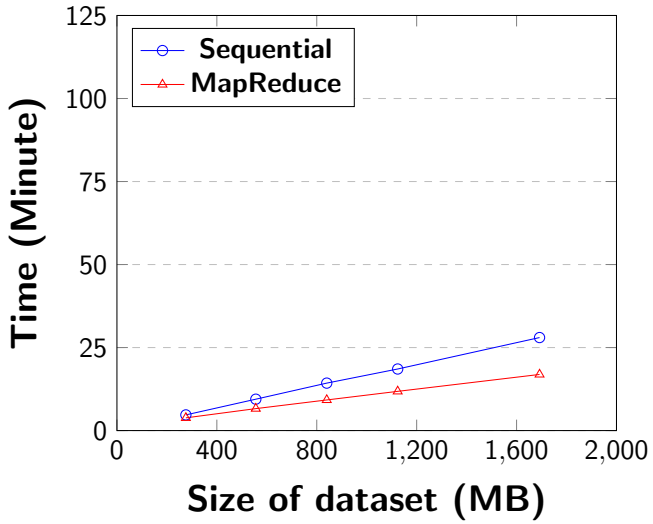
No. of Tuples	**Size in MB**	**Sequential**		**MapReduce**		
		T_g	T_d	M	T_g	T_d
5810120	276	3.89	4.72	2	3.52	3.83
11620240	556	7.77	9.45	5	5.74	6.61
17430360	840	11.13	14.29	7	8.24	9.22
23240480	1124	15.28	18.54	9	10.95	11.81
34860720	1692	22.38	28.01	13	15.51	16.90

Table: Results of sequential and MapReduce version for G = 50000

Experimental Results: [Bhattacharya and Cortesi, 2009]



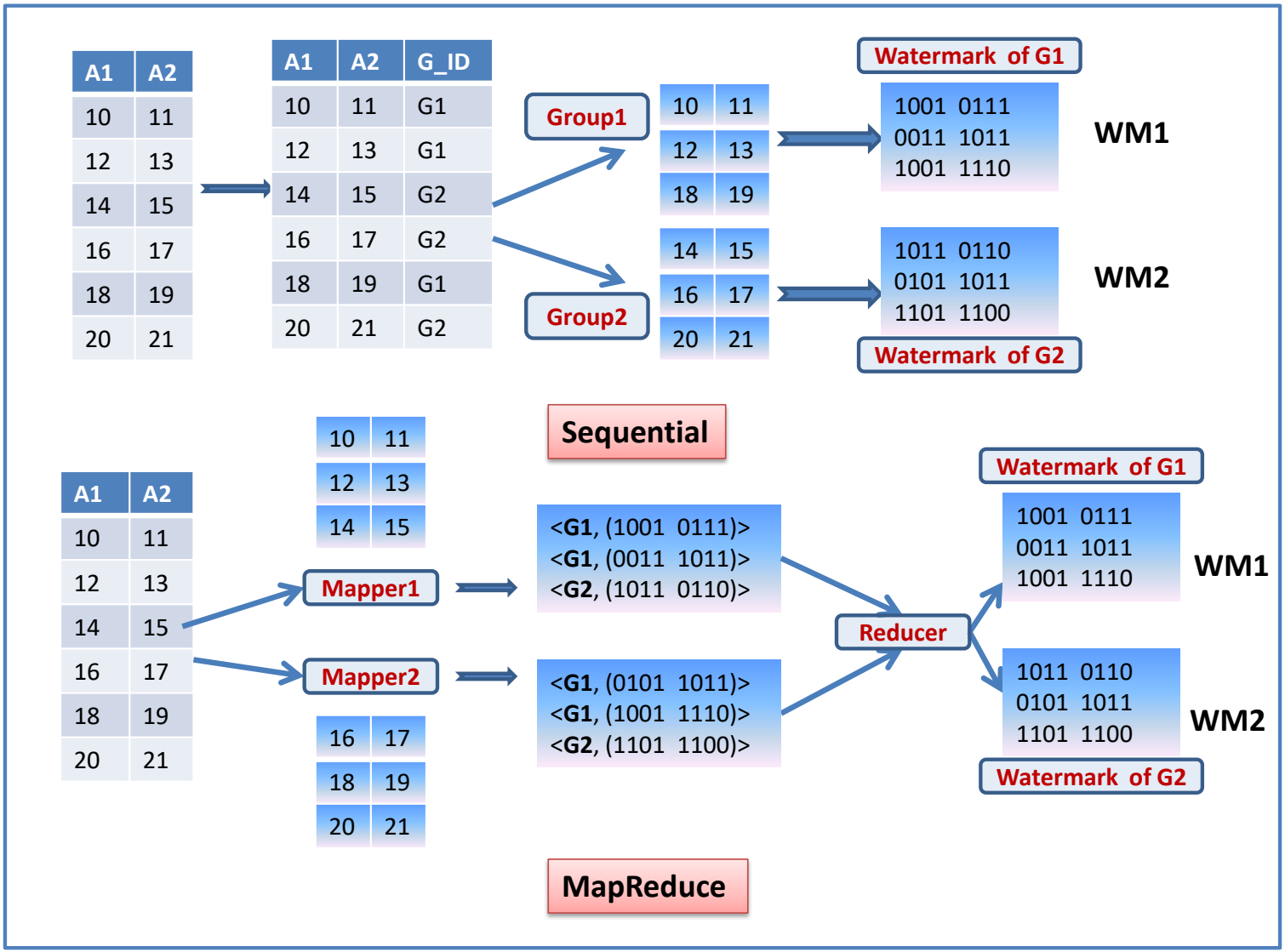
(a) Watermark Generation Time



(b) Watermark Detection Time

Figure: Performance comparison of sequential and MapReduce for G = 50000

Experimental Results: [Bhattacharya and Cortesi, 2010]

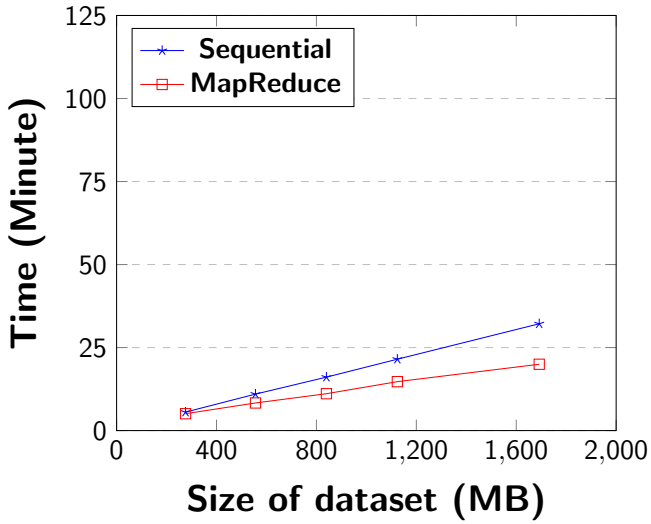


Experimental Results: [Bhattacharya and Cortesi, 2010]

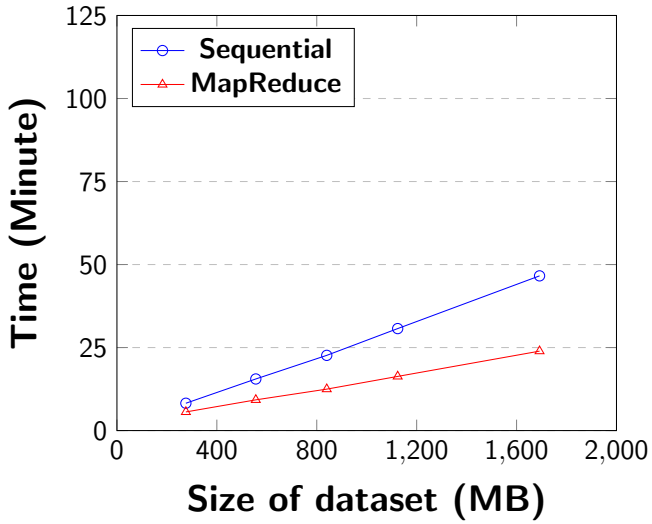
No. of Tuples	Size in MB	Sequential		MapReduce		
		T_g	T_d	M	T_g	T_d
5810120	276	5.56	8.23	2	5.08	5.61
11620240	556	10.98	15.54	5	8.31	9.25
17430360	840	16.09	22.65	7	11.10	12.48
23240480	1124	21.49	30.71	9	14.73	16.31
34860720	1692	32.16	46.59	13	19.99	23.92

Table: Results of sequential and MapReduce version for G = 50000

Experimental Results: [Bhattacharya and Cortesi, 2010]



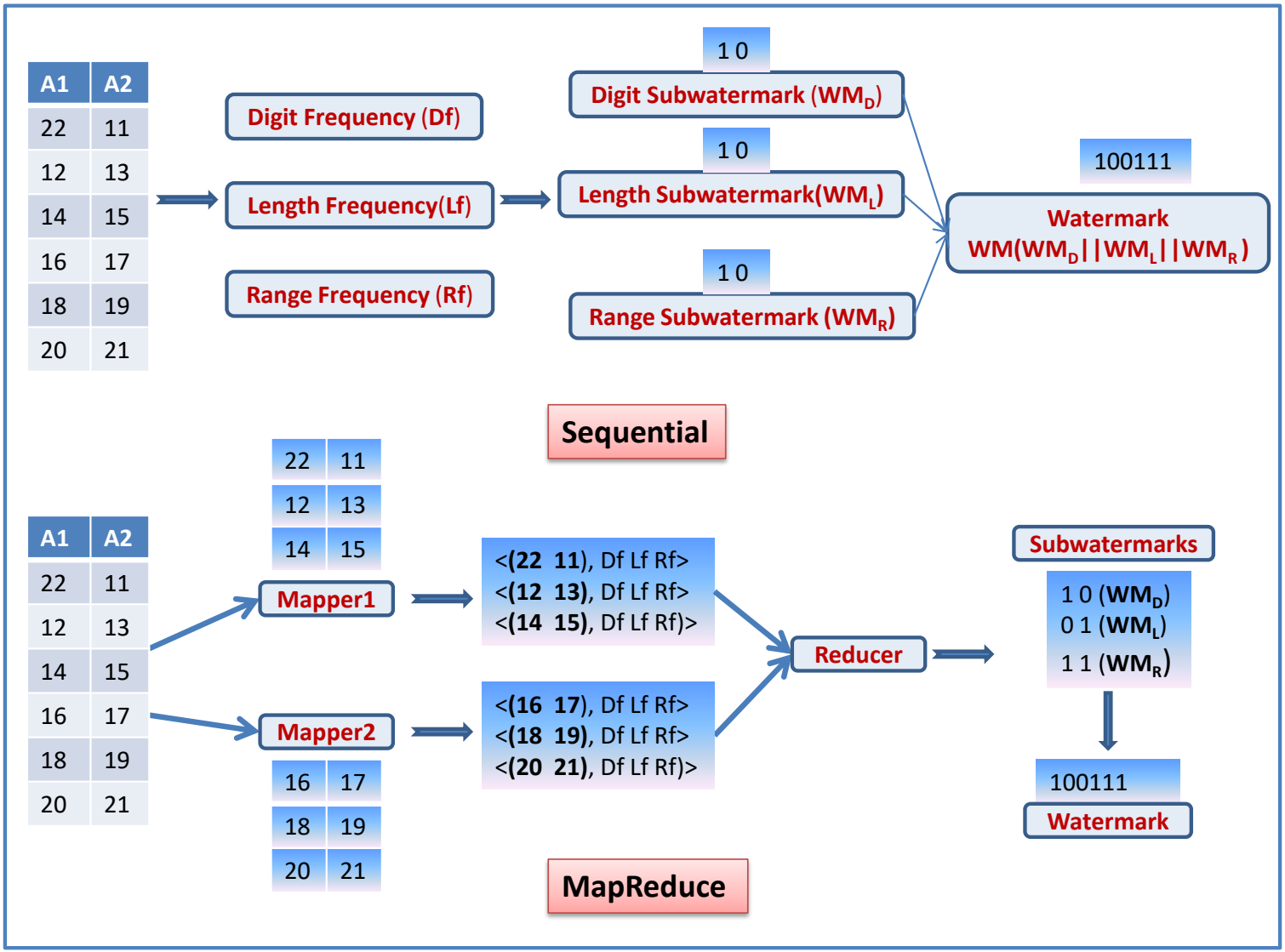
(a) Watermark generation time



(b) Watermark detection time

Figure: Performance comparison of sequential and MapReduce for G = 50000

Experimental Results: [Khan and Husain, 2013]

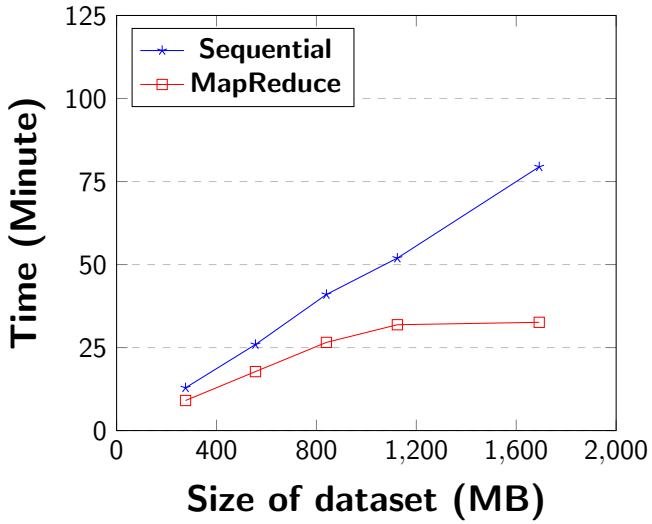


Experimental Results: [Khan and Husain, 2013]

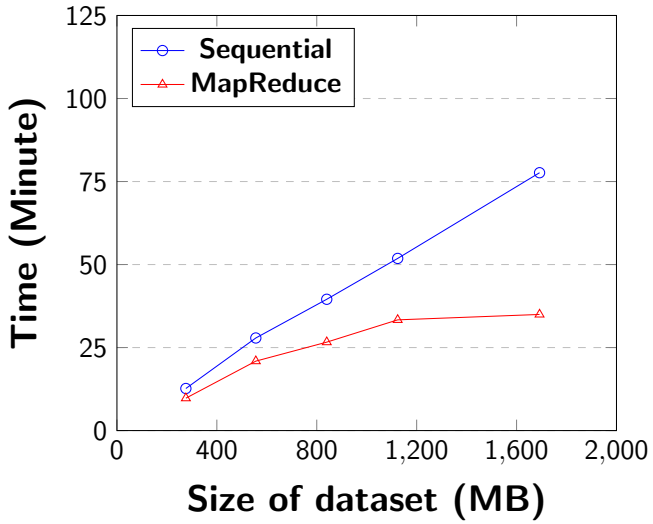
No. of Tuples	Size in MB	Sequential		MapReduce		
		T_g	T_d	M	T_g	T_d
5810120	276	12.88	12.67	2	9.05	9.75
11620240	556	25.91	27.88	5	17.76	20.91
17430360	840	40.98	39.54	7	26.55	26.63
23240480	1124	51.93	51.84	9	31.87	33.34
34860720	1692	79.43	77.65	13	32.60	34.97

Table: Results of sequential and MapReduce version for G = 50000

Experimental Results: [Khan and Husain, 2013]



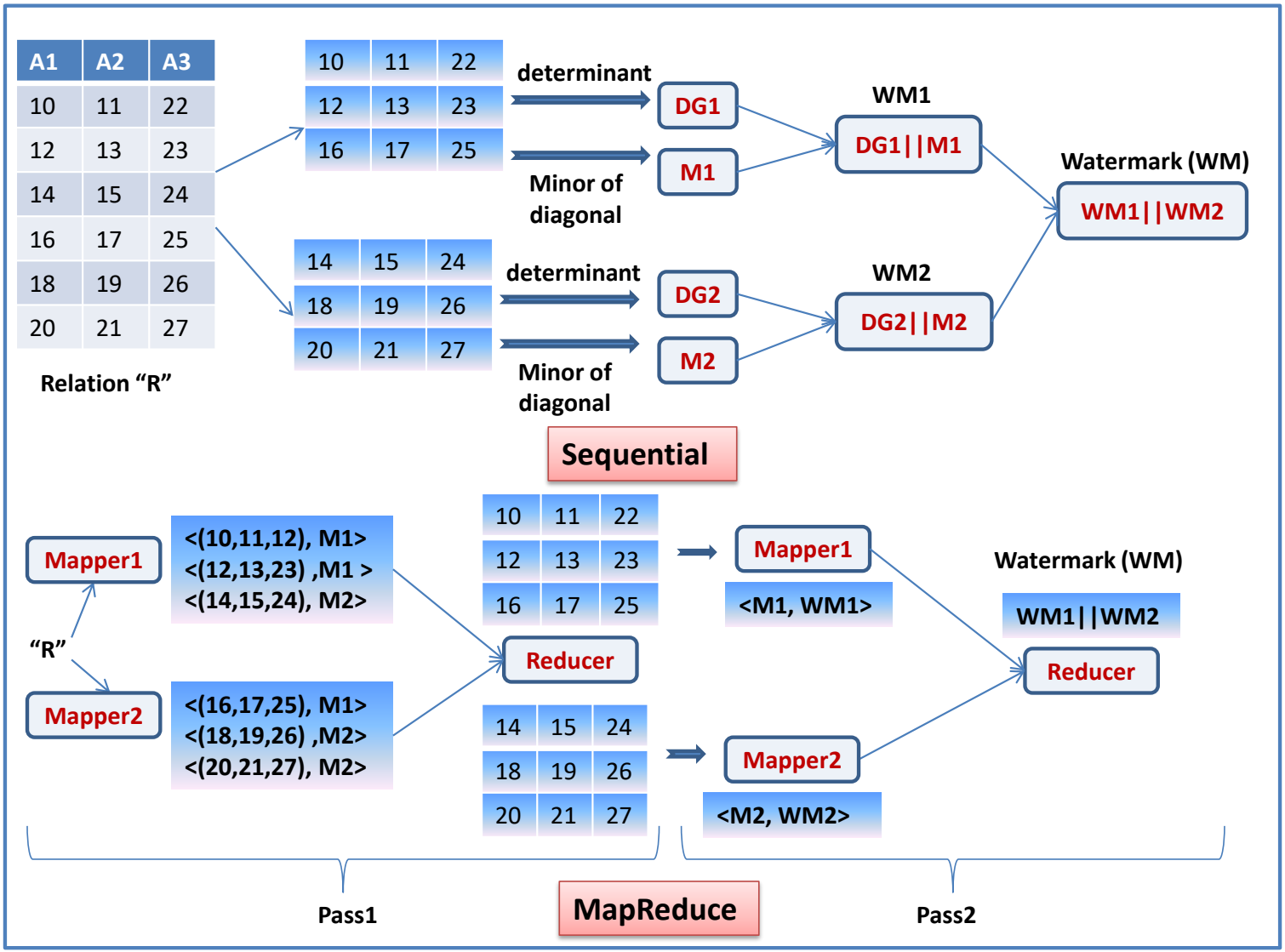
(a) Watermark generation time



(b) Watermark detection time

Figure: Performance comparison of sequential and MapReduce for G = 50000

Experimental Results: [Camara et al., 2014]

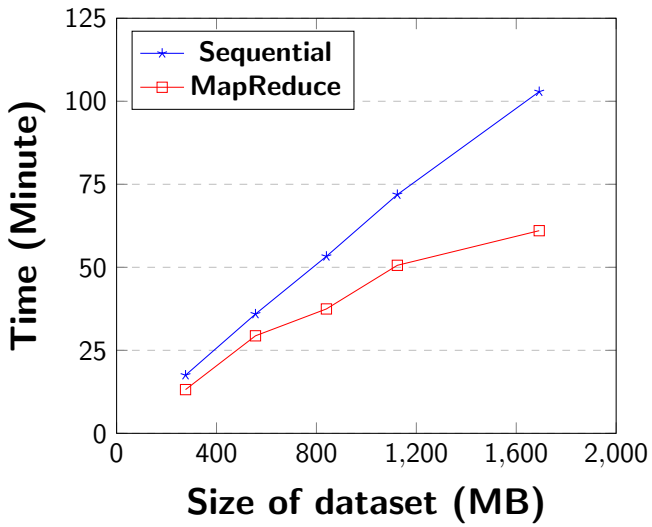


Experimental Results: [Camara et al., 2014]

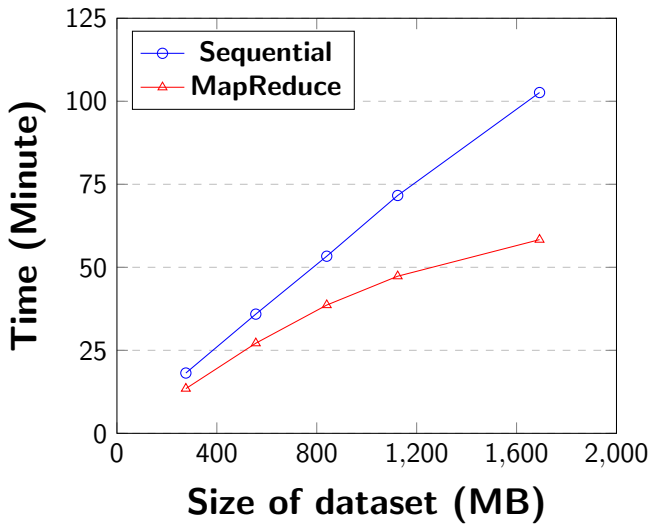
No. of Tuples	Size in MB	Sequential		MapReduce		
		T_g	T_d	M	T_g	T_d
115	276	17.52	18.16	2	14.40	13.47
230	556	35.92	35.90	5	29.37	27.14
345	840	53.30	53.34	7	37.46	38.63
460	1124	71.90	71.63	9	50.60	47.32
690	1692	102.89	102.63	13	61.03	58.33

Table: Results of sequential and MapReduce Version for G = 50000

Experimental Results: [Camara et al., 2014]



(a) Watermark generation time



(b) Watermark detection time

Figure: Performance comparison of sequential MapReduce for G = 50000

Experimental Results: Percentage of time-reduction

- t_s = time required in Sequential
- t_m = time required in MapReduce-based approach.
- % of time-reduction= $\frac{t_s - t_m}{t_s} \times 100$.

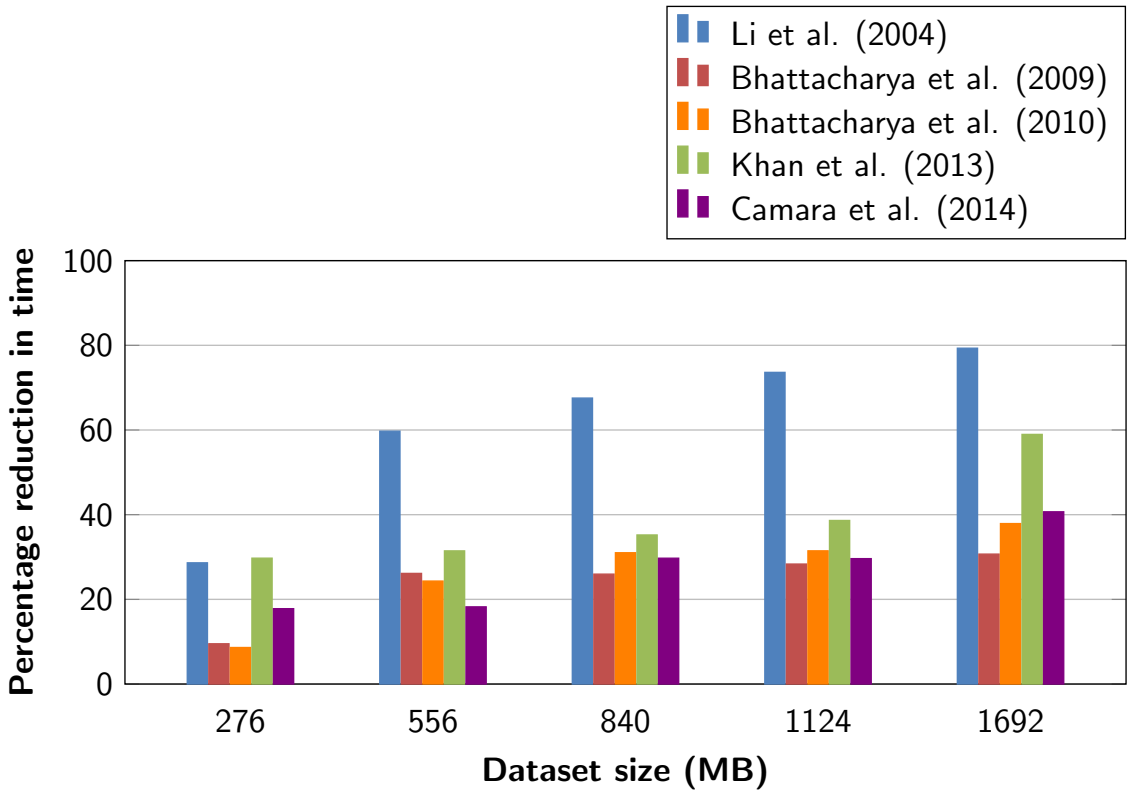


Figure: Watermark generation

Experimental Results: Percentage of time-reduction

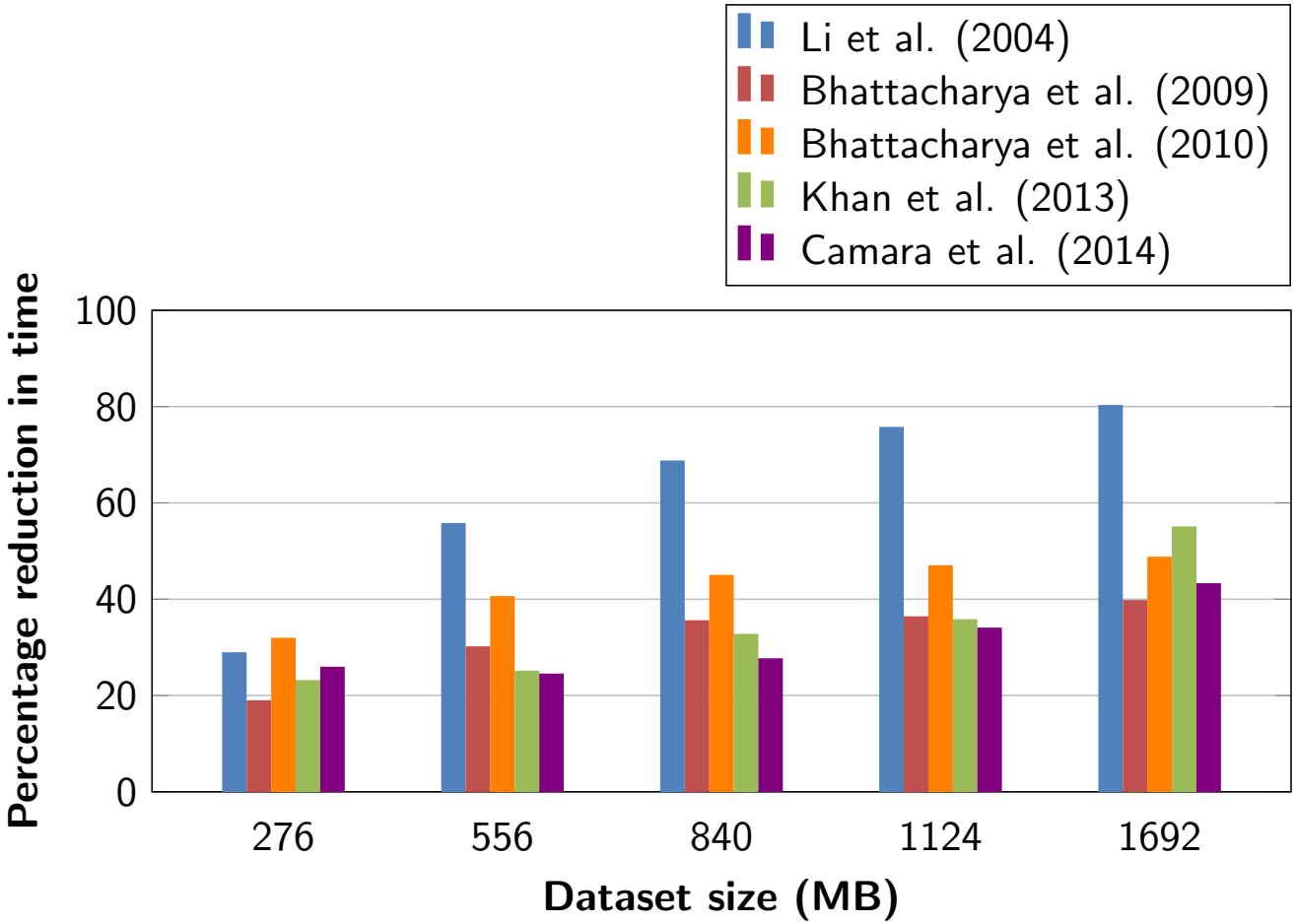


Figure: Watermark detection

Common observations

- The watermark generation and detection time reduce significantly in MapReduce as compared to their sequential executions.
- Performance improves as the number of mappers increases.
- % of time-reduction in watermark generation and detection increases as we increase the size of the dataset.

Outline

Conclusions and Future Works

- We have focused on distortion-free watermarking
- Significant reduction in watermark generation as well as detection time in MapReduce.
- Percentage reduction of time from sequential to MapReduce increases with the increase of dataset size.
- To the best of our knowledge, this is the first work towards MapReduce-based big relational database watermarking.
- **Future Work** - Big Data watermarking.

Outline

References I

[Agrawal and Kiernan, 2002] Agrawal, R. and Kiernan, J. (2002).
Watermarking relational databases.
In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 155–166. VLDB Endowment.

[Bhattacharya and Cortesi, 2009] Bhattacharya, S. and Cortesi, A. (2009).
A generic distortion free watermarking technique for relational databases.
In *International Conference on Information Systems Security*, pages 252–264. Springer.

[Bhattacharya and Cortesi, 2010] Bhattacharya, S. and Cortesi, A. (2010).
Distortion-free authentication watermarking.
In *International Conference on Software and Data Technologies*, pages 205–219. Springer.

[Camara et al., 2014] Camara, L., Li, J., Li, R., and Xie, W. (2014).
Distortion-free watermarking approach for relational database integrity checking.
Mathematical problems in engineering, 2014.

[Dean and Ghemawat, 2008] Dean, J. and Ghemawat, S. (2008).
Mapreduce: simplified data processing on large clusters.
Communications of the ACM, 51(1):107–113.

References II

[Guo et al., 2006] Guo, H., Li, Y., Liu, A., and Jajodia, S. (2006).
A fragile watermarking scheme for detecting malicious modifications of database relations.
Information Sciences, 176(10):1350–1378.

[Halder et al., 2010] Halder, R., Pal, S., and Cortesi, A. (2010).
Watermarking techniques for relational databases: Survey, classification and comparison.
J. UCS, 16(21):3164–3190.

[Kamran et al., 2013] Kamran, M., Suhail, S., and Farooq, M. (2013).
A robust, distortion minimizing technique for watermarking relational databases using once-for-all usability constraints.
IEEE Transactions on Knowledge and Data Engineering, 25(12):2694–2707.

[Khan and Husain, 2013] Khan, A. and Husain, S. A. (2013).
A fragile zero watermarking scheme to detect and characterize malicious modifications in database relations.
The Scientific World Journal, 2013.

[Li et al., 2004] Li, Y., Guo, H., and Jajodia, S. (2004).
Tamper detection and localization for categorical data using fragile watermarks.
In *Proceedings of the 4th ACM workshop on Digital rights management*, pages 73–82. ACM.

References III

[Pournaghshband, 2008] Pournaghshband, V. (2008).
A new watermarking approach for relational data.
In *Proceedings of the 46th annual southeast regional conference on XX*, pages 127–131. ACM.

[Raghupathi and Raghupathi, 2014] Raghupathi, W. and Raghupathi, V. (2014).
Big data analytics in healthcare: promise and potential.
Health Information Science and Systems, 2(1):3.

[Rani et al., 2016a] Rani, S., Kachhap, P., and Halder, R. (2016a).
Data-flow analysis-based approach of database watermarking.
In *Advanced Computing and Systems for Security*, pages 153–171. Springer.

[Rani et al., 2016b] Rani, S., Koshley, D. K., and Halder, R. (2016b).
A watermarking framework for outsourced and distributed relational databases.
In *International Conference on Future Data and Security Engineering*, pages 175–188. Springer.

[Russom, 2013] Russom, P. (2013).
Managing big data.
TDWI Best Practices Report, TDWI Research, pages 1–40.

Thank You
For Your Kind Attention

Any suggestions please...