International Journal of Computer Applications (0975 – 8887)
Volume 109 – No. 5, January 2015

A Cyber-Physical Stream Algorithm for Intelligent Software Defined Storage

Adi Alhudhaif Maryam Yammahi Tong Yan Simon Berkovich
Department of Computer Science, The George Washington University
Washington, DC. 20052, USA

ABSTRACT

The paper presents a new Cyber Physical Stream algorithm for selecting a predominant item from very large collections of data. The algorithm effectively works for frequencies of the predominant items starting from about 2%. The algorithm is focused on querying massive data in Software-Defined Storage combined with Fuzzy indexing method. Experiment results show that Cyber Physical Stream algorithm improves the accuracy and efficiency over previous efforts.

Keywords

Big Data processing; Stream Algorithms; Software-Defined Storage; Majority algorithm; Fuzzy search; Database Management Systems; Pigeonhole principle.

1. INTRODUCTION

Software-Defined Storage (SDS) is a data storage infrastructure that allows software plays a significant role instead of merely relying on storage hardware. Nowadays, the increasing data amount and complexity propose challenges for Data Storage Center. The widely used Software-Based Storage can deal with most of demands [1]. In the Software-Defined Storage, the software would not only take the responsibility of Data Allocation, Data Management, Data Maintenance and etc., by a series of computing, Software-Defined Storage also creates a more automatic system. In the end, Software-Defined Storage will provide more efficient, flexible and intelligent storage services. The Software-Defined Storage is still in the stage of developing. The Software-Defined Storage has features including API support, data visualization and hybrid cloud storage [2]. Software-Defined Storage can improve the usage of existing storage assets so that fewer new devices need to be purchased. The idea is to create a single pool of logical storage from a number of devices. Storage administrators can then provision that pool of storage as needed, thereby improving the use of storage arrays and making the whole affair easier to manage[3]. This new model of Software-Defined Storage is cost effective that facilitates large datacentre customers to acquire high performance storage at lower costs than when using traditional storage vendors [4].

In this paper, we emphasize on the role of stream processing in a specific Software-Defined Storge that is introduced by work [5]. In section 2, we give a short overview of the suggested Software-Defiend Storage. In section 3, we intrduce a novel stream algorithm called Cyber-Physical Stream (CPS) that selects the predominant item from very large collections of data. Moreover, we introduce the algorithm of CPS and physical design that holds CPS. Section 3.2 covers the role of CPS and other majority algorithm presented in previous work [6]. In section 4, we present the results of our experiments on CPS with input of stream data in random uniform distribution and stream data follows Zipf’s law distribution.

2. AN INTELLIGENT SOFTWARE DEFINED STORAGE

The Big Data situation requires a qualitatively different type of information processing. This problem brings in a new type of a computational model that explicitly works only with a relatively small portion of the available data, while the rest of the data just implicitly affects selection of the given working portion [7]. The unavoidable restrictions on the operations with overabundant data translate into the design of the brain in accordance with the fundamental Freud’s idea of unconsciousness. This design is contemplated in our paper [8]. Diversified information in overwhelming amounts appears ambiguous, volatile, and unreliable. So, the contents of Big Data systems cannot be treated with confidence as in traditional searching and data mining. Instead, Big Data should be utilized essentially through what can be seen as “knowledge formation”. In other words, processing of Big Data must be performed by what can be considered as “scientific method”. Namely, besides simple extraction of references as from regular information systems the full exploitation of Big Data necessitates formulating testable hypotheses and creating prediction models. A classical illustration presents usage of the observational data of Tycho Brahe through transformation of Kepler’s laws into Newton’s model of “Universal Gravitation”. Thus, employing Big Data falls into the realm of Artificial Intelligence. As a matter of fact, the intelligence facilities of the brain can be considered as a necessary condition to deal with the Big Data challenge. A special type of holographic memory is a pivot point in the realization of these facilities [8].

To implement such kind of Big Data processing facilities in practice we introduce a particular construction of intelligent Software-Defined Storage (SDS). Software-Defined Storage is the term for data storage technology that separates the hardware storage from the software that manages the storage infrastructure. This way the storage infrastructure resources can be automatically and efficiently be allocated and managed to fulfill the enterprise’s need. The construction of the suggested Software-Defined Storage emulates the basic features of the suggested memory organization of the brain: multi-attribute cortical map, content-addressable access, and stream resolution of multiple responses. The envisioned Software-Defined Storage incorporates two developments: memory device for multi-attribute items that can be accessed by any combinations of attributes using FuzzyFind procedures [9] and massive distributed streaming for resolution of multiple responses [6].

The suggested storage accumulates various information items from outside at arbitrary rates. Each item contains number of attributes. Attributes of information items are characterized by 23-bit metaknowledge templates [7], thus the organized items will have less number of attributes. The enterprise’s request is a set of some number of specified attributes. Access to storage issues a request from different component. Content-addressable access is arranged by inverted files for each type of attribute using the FuzzyFind Dictionary (FFD) with Pigeonhole search algorithm introduced in[9]. The result of this request marks a large subset of various data items. Accessing these items with individual attributes, we will get a lot of responses (multiple responses), especially if we access approximately with the suggested Pigeonhole method using FuzzyFind Dictionaries in [9]. A stream algorithm is used for the resolution of muiltiple responses. Appling that to the marked data items, does the selection of appropriate items. We will formulate certain common criterion for retrieval, and perform the extraction of an element in a stream fashion. This paper focuses in the used stream algorithm; therefore, the next sections will include more details about it.

Generally speaking, we have a lot of attributes, and to make inverted FuzzyFind dictionaries for each attribute is expensive. For these attributes, Search is more rare and can be done by sequential streaming of the whole storage. Therefore, We have used parts of the attributes only to access in our design, and this is based on the newly discovered switch in the brain consciousness [10]. This will substantially simplify the construction. In our design, we select Primary attributes on which we will perform access search, and this where we use the FuzzyFind Dictionary for access. The Secondary attributes, at which only the selection of the retrieved attributes is to be performed, and this where we use the stream algorithm.

3. CYBER-PHYSICAL STREAM ALGORITHM

In the field of computer science, stream algorithms are being designed for processing data streams with limitations of processing time and memory. In 1999, streaming algorithms were introduced [11][12], and used in most fields of computer science such as; networking [13], machine learning [14], information security [15], web application, manufacturing, financial applications, telecommunications data management, and much more. Using traditional database systems by simply inserting the incoming stream data, and process them locally is not a feasible solution and it is not a best practice solution. Traditional database management systems are not structured for *continuous queries* that are typical of data stream applications [16].

Finding the most frequent item is considered one of the most heavily studied problems since 1980s[17], and it is one the major problems in data stream processing [18][19]. Also, many domains of computer science such as computer networks, data mining and database would benefit from finding the most frequent item [20] [21] [22]. In section 3.1 we first introduce a novel Cyber-Physical Stream (CPS) algorithm that has very high probabilities of selecting the most frequent item of frequencies as low as 2%. Moreover, we introduce a design of an analog physical device that holds the CPS algorithm that extracts prevalent items from streams of big data. In section 3.2 we show the major role of the CPS algorithm and Multi-Buffer Based algorithm [6] in the suggested structure of Software-Defined Storage [5] in more details.

3.1 The Design of CPS Algorithm.

Cyber-Physical Stream (CPS) algorithm is a novel algorithm that in very high probabilities, it extracts the item of most frequent occurrence of frequencies as low as 2%. The CPS algorithm was inspired by the possibility of its usage in the model of the brain as considered in [8]. The algorithm is structured as a physical process that can be illustrated in Fig*ure* .1.

Cyber-Physical Stream algorithm is stated as the following: initially assign the value 1 to Weight (T), and the value 0 to Voltage Control (v). For every coming stream slice do the following: store the new arrival of the stream slice and set Voltage Control (v) to 1. Keep comparing the incoming stream slice with previously stored item. If they are same, and Voltage Control is greater than or equal to 1, increase the Voltage Control (v) by the result value of $\frac{T}{v}$, and increase the Weight (T) by 1. However, if not the same, and Voltage Control (v) is greater than 1, decrease the Voltage Control (v) by a decrement rate known by "α". Otherwise, next new arrival of stream slice will replace the previous stored item and set Voltage Control to 1. When the last slice of the stream has processed, the stored item is prevalent item. Cyber-Physical Stream algorithm is described as follows:

```
%% Initial values
T = 1;
v = 0;
%% Stream started
REPEAT
get next item
If (v >= 1 and item == top):
     v = v + ( T / v )
     T = T + 1
Else_if (v > 1 and item ≠ top):
     v = v * α
Else:
     v = 1
     top = item
UNTIL more items
%% Stream finished
```

Fig*ure*. 1, presents a physical design for an analog device that holds the CPS algorithm that extracts the prevalent item from streams of big data. In more detail, at the time of new slice of stream arrived to the Reception Management, it sends a signal to voltage management. If the new incoming item is the same as the one in reception register, voltage control increases voltage. If not the same, it does nothing, and voltage decreases exponentially.

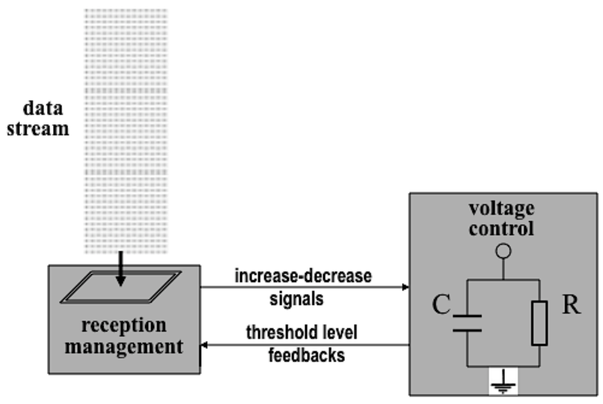

Figure 1. Illustration of the scheme for extracting the prevalent item from a stream

3.2 The Functionality of CPS and Multi-Buffer Based Algorithms in intelligent Software-Defined Storage

In previous work [6], we introduced Multi-Buffer Based algorithm that has the ability of retrieving the top k most frequent items in big data stream in leaner time O(n) with very limited space and memory requirements. In this paper we introduced the CPS that also has the time complexity of O(n), and limited space and memory requirement to extract the most frequent item over a stream of big data.

Following the suggested structure of the intelligent Software Defined Storage, using stream processing represents the second half of the organization. Either the CPS or Multi-Buffer Based algorithms represent stream processing in our case. The stream algorithms are used for the resolution of multiple responses that are the outputs of using inverted FuzzyFind Dictionaries as mentioned previously. By accessing multi-attribute items using any combination of attributes, it produces a vast amount of data to be processed in stream processing fashion to select the most appropriate items based on their occurrence frequencies.

4. CPS EXPERIMENTS/RESULTS

4.1 Setup

Experiments are divided into two main parts: Firstly, performing Cyber-physical algorithm using streams of uniform random numbers. Secondly, performing the CPS algorithm using streams follow zipf's law distribution. For every single stream with predefined element frequency we created many iterations using The Fisher-Yates shuffle algorithm [23][24]. Generating uniform random numbers was performed using both generator functions in Python’s library Lib/random.py and the random number libraries in C that takes variable seeds such as: current system time to generate uniform random numbers. For generating numbers followed Zip’f Law, we first use the same shuffle algorithm to generate uniform random data. According to Zip’f Law Distribution, we specify a range of numbers to the most frequent element. Then, according to the most frequent element, we specify a range of numbers to the second most frequent element. And the rest are generated by the frequent of last element. We performed and examined Cyber physical algorithm using both kinds of streams under a common implementation framework to test their performance as accurately as possible. The algorithm was implemented using both C and Python, and compiled using gcc on Cygwin 1.7.25 for C code, and Python 2.7.5 for python code. We ran Python experiments on 2.6GHz dual-core Intel Core i5 with 8GB of RAM running OS X 10.9.2. Experiments of algorithms in C were ran on Intel 4th generation core i5 using 8GB of RAM running Microsoft Windows Server 2012. We did not observe noteworthy differences between two compilers.

4.2 Uniform Random Numbers

After many expermints of the CPS using streams of uniform random numbers and many of its iterations as an input, we came to the restult that the value of α = 0.99 will get the best results out of the CPS. Each stream is a size of 1000 items that contain an item of predefined frequency. Fig*ure.* 2. Illustrates the probabilities of retrieving the prevalent item with various frequencies in α = 0.99.

Fig*ure*. 3 show the factor of stream size, we performed the CPS algorithm using α = 0.99 over three different stream sizes as follows: 1,000, 10,000 and 100,000. Frequencies tested are 1%, 2%, 3%, 4% and 5%. Therefore, increasing stream size will enhance the performance of the CPS algorithm.

4.3 Zipf’s Law Distribution

For stream data follows Zipf’s law distribution, it was observed that α = 0.8 provides the best performance out of the CPS algorithm. In this section, we performed CPS algorithm using α = 0.8 and stream input that follows Zipf’s law distribution. Figure. 4 illustrate the probabilities of retrieving the first most frequent item and the second most frequent item.

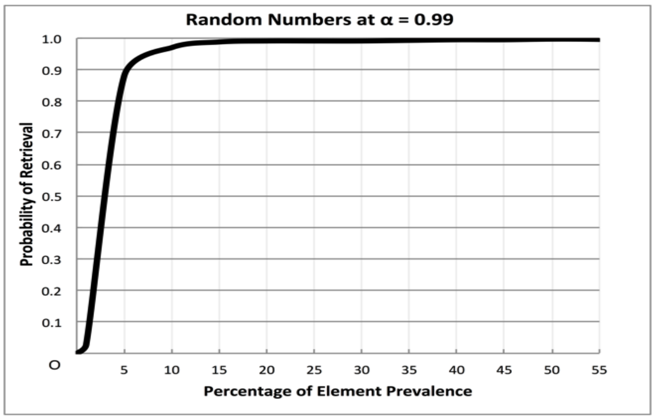

Figure 2. Probabilities of retrieving the prevalent item

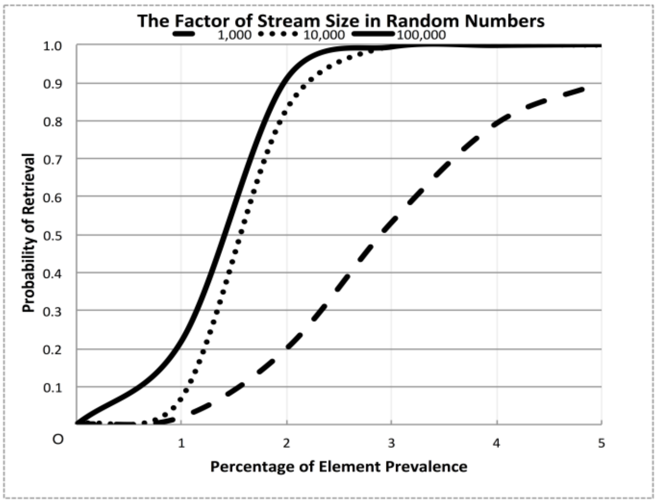

Figure 3. Probabilities of retrieving the prevalent item in various stream sizes

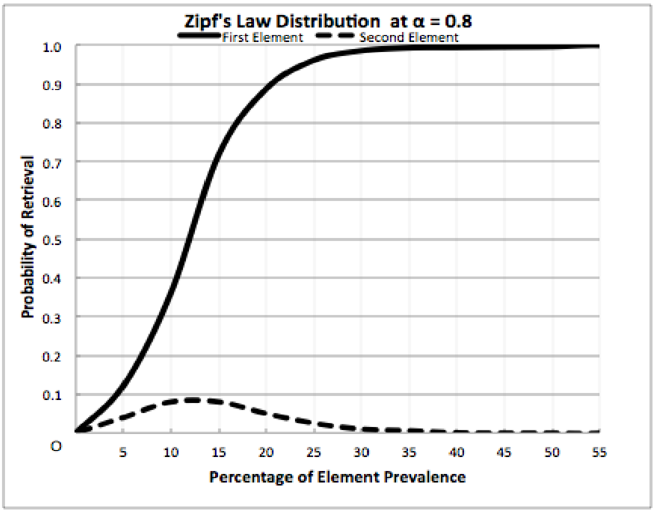

Figure 4. Probabilities of retrieving first and second most frequent item

For stream data that follows Zipf's law distribution, Figure. 5 show the influence of stream size over the CPS algorithm. The input size that has 200,000 numbers has better performance than the one that has 10,000 elements. However, compared to the improvement in random distribution, the improvement here is minimal. The reason for this minimal improvement is the existence of the second prevalent element in the Zipf's Law Distribution stream. The accumulation impacts would increase the influence of the second prevalent element. Even the most prevalent element would have more accumulation; however, the increasing influence of the second prevalent element would offset some accumulation of the prevalent element.

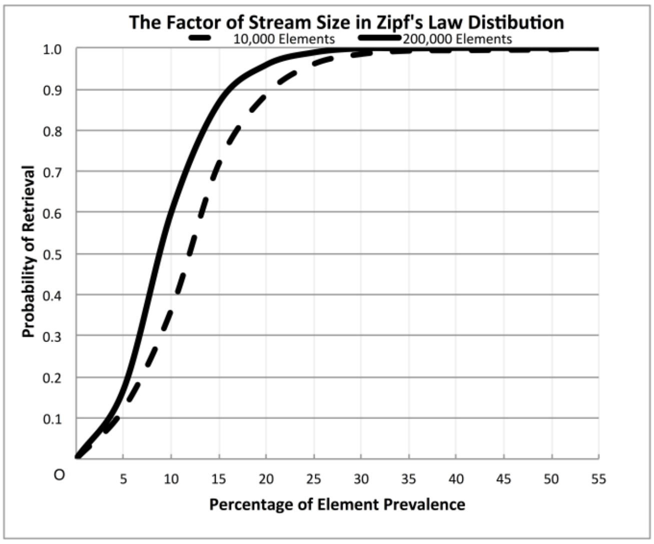

Figure 5. Probabilities of retrieving the prevalent item in different stream sizes

5. REMARKS/CONCLUSIONS

The performance of the algorithm is determined by the relationship between increment and decrement rates. For random distribution determining a certain increment rate, the optimal decrement rate turns out to be presented by $\alpha = 0.99$. In Fig*ure*. 6, we show the improvement of the CPS algorithm compared to previous work [6] Single-Buffer Based algorithm. The CPS algorithm performs better at providing very high probabilities of selecting the predominant item of frequencies as low as 2%. Fig*ure*. 7 illustrate the comparison of the CPS algorithm and our previous work Single-Buffer Based algorithm [6] using data streams follow Zipf's law distribution. It was observed that the new CPS algorithm has better performance of providing high probabilities of retrieving the prevalent item of low frequencies. For Zipf's law, the decrement rate that is optimal for a selection of a single element is $\alpha = 0.8$. However, Zipf's law is intended for selection of a group of elements. Thus, selecting two most frequent elements in accordance with Zipf's law using $\alpha = 0.99$ may provide more better results if applied to pairs of prevalent elements. Anyhow, single Zipf's distribution selection does not have to be optimal for $\alpha = 0.99$ as does random distribution selection.

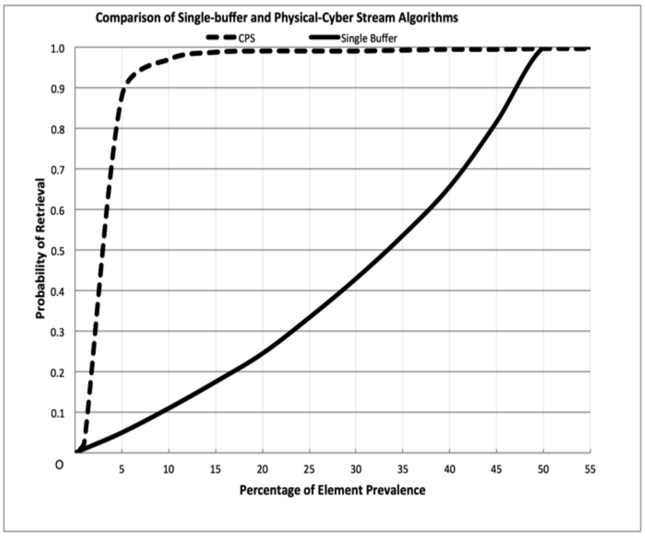

Figure 6. Comparison of Single-Buffer Based and CPS Algorithms using streams of uniform random items

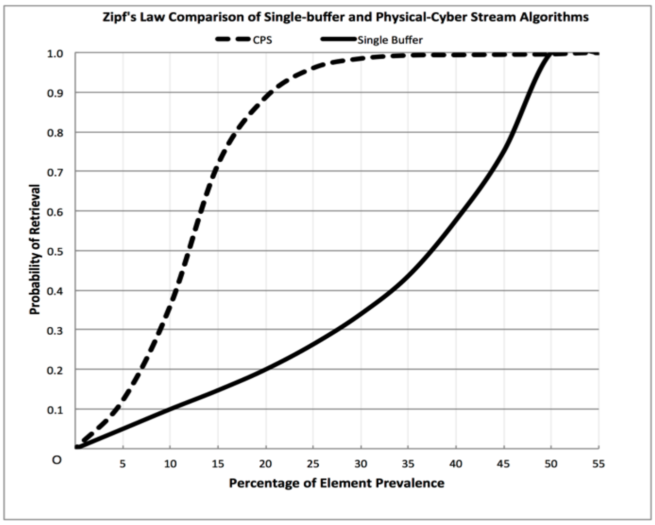

Figure 7. Comparison of Single-Buffer Based and CPS Algorithm using streams of data follow Zipf's law distribution

6. REFRENCES

[1] "Understanding the DNA of Software Defined Storage", http://www.vmware.com/files/pdf/solutions/Understanding-the-DNA-of-Software-Defined-Storage-Tech-trends.

[2] "IBM software defined storage," http://www-03.ibm.com/systems/storage/software-defined-storage

[3] Moore, John. "Software-defined Storage Aims for Ease of Management." Software-defined Storage Aims for Ease of Management -- FCW. May 23, 2014. Accessed October 12, 2014.

[4] "Australia : SOFTWARE Defined Storage Introduced by CloudCentral." 2014. MENA Report. http://search.proquest.com/docview/1516630033?accountid=11243

[5] Simon Berkovich, "Intelligent Software Defined Storage", in Proceedings of the 5th international Conference on Computing for Geospatial Research & Application, Washington, D.C., 2014.COM.Geo

[6] Adi Alhudhaif, Tong Yan and Simon Berkovich. “On the organization of cluster voting with massive distributed streams”, in Proceedings of the 5th international Conference on Computing for Geospatial Research & Application, Washington, D.C., 2014.COM.Geo

[7] Simon Berkovich, Duoduo Liao, “On clusterization of big data streams”, COM.Geo '12 Proc. of the 3rd International Conference on Computing for Geospatial Research and Applications, ACM, New York, 2012

[8] Simon Berkovich, “”Organization of the Brain in Light of the Big Data Philosophy”, COM. BigData’ 14 Proc. of the 1st International Summits on Big Data Computing, IEEE, Washington DC, 2014.

[9] M. Yammahi, K. Kowsari, Chen. Shen and Simon Berkovich, "An efficient technique for searching very large files with fuzzy criteria using the Pigeonhole Principle”, COM. BigData’ 14 Proc. of the 1st International Summits on Big Data Computing, IEEE, Washington DC, 2014.

[10] H. Thomason, “ Consciousness on-off switch discovered deep in brain”, New Scientist. Retrieved from http://www.newscientist.com/article/mg22329762.700-consciousness-onoff-switch-discovered-deep-inbrain.html#.U80nJBbZXdl

[11] Alon, Noga, Yossi Matias, and Mario Szegedy. "The space complexityof approximating the frequency moments." Proceedings of the 28thannual ACM symposium on Theory of computing. ACM, 1996

[12] Babcock, Brian; Babu, Shivnath; Datar, Mayur; Motwani, Rajeev;Widom, Jennifer (2002), "Models and issues in data stream systems",Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposiumon Principles of Database Systems (PODS 2002), pp. 1–16,doi:10.1145/543613.543615

[13] Abadi, Daniel J., et al. "The Design of the Borealis Stream Processing Engine." CIDR. Vol. 5. 2005.

[14] Rosten, Edward, and Tom Drummond. "Machine learning for highspeed corner detection." Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006. 430-443.

[15] N. Gruschka, M. Jensen, L. Iacono, and N. Luttenberger, “Server-side streaming processing of WS-Security,” IEEE Transactions on Services Computing, vol. 4, no. 4, pp. 272–285, 2011.

[16] D. Terry, D. Goldberg, D. Nichols, and B. Oki. Continuous queries over append-only databases. In Proc. of the 1992 ACM SIGMOD Intl. Conf. on Management of Data, pages 321–330, June 1992.

[17] Boyer, R.S., Moore, J.S. A fast majority vote algorithm. Technical Report ICSCA-CMP-32, Institute for Computer Science, University ofTexas (Feb. 1981)

[18] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. Journal of Computer and System Sciences, 58(1):137–147, 1999.

[19] Monika Henzinger, Prabhakar Raghavan, and Sridhar Rajagopalan. Computing on data streams. Technical Report SRC TR 1998-011, DEC, 1998.

[20] M. Fang, N. Shivakumar, H. Garcia-Molina,R. Motwani, and J. D. Ullman. Computing icebergqueries efficiently. In Proc. of VLDB, 1998.

[21] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. of VLDB, 1994.

[22] C. Estan and G. Varghese. New directions in trafficmeasurement and accounting. In Proc. of ACMSIGCOMM, 2002.

[23] Richard Durstenfeld, Algorithm 235: Random permutation, Communications of the ACM, v.7 n.7, p.420, July 1964.

[24] Fisher, Ronald A.; Yates, Frank (1948) [1938]. Statistical tables for biological, agricultural and medical research (3rd ed.). London: Oliver & Boyd. pp. 26–27.