MASTERARBEIT

Organisation von Bloom-Filtern zur effizienten k-nächste-Nachbarn-Suche in einem kontextzentrischen sozialen Netz

Judith Greif

MASTERARBEIT

Organisation von Bloom-Filtern zur effizienten k-nächste-Nachbarn-Suche in einem kontextzentrischen sozialen Netz

Judith Greif

Aufgabenstellerin:	Prof. Dr. Claudia Linnhoff-Popien
Betreuer:	Mirco Schönfeld
	Dr. Martin Werner
Abgabetermin:	26. Juli 2016

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 26. Juli 2016

....
(Unterschrift der Kandidatin)

Abstract

Hier kommt der Abstract hin. Hier kommt der Abstract hin.

Inhaltsverzeichnis

1 Einleitung

Referenzen bis jetzt: [AT06], [ADI$^+$12], [BMS07], [BM04], [BCM02], [DWM10], [HP94], [LC86], [Naf05], [QLC14], [RK14], [SBE$^+$12], [Sch13], [SW14], [STT$^+$09], [WDS15], [YL02], [Zha12], [ZJW04], [Jan95].

1.1 Problembeschreibung

2 Hintergrund

2.1 Bloom-Filter

2.1.1 Grundlagen

Attenuated Bloom Filter: [SS11]: 316 and 318
Counting Bloom Filter: [FCAB00]
Compressed Bloom Filter: [Mit02]

2.1.2 Mathematische Prinzipien und Operationen

2.2 Indexstrukturen

To support query processing and operations in an efficient manner, the internal layer of a database system uses specific data strucures and memory methods. These are called *index structures*. They organize the data to support the required operations using its *indices*.

An *index* (also called *directory*) of a file holds information about its structure. A *file* in this context refers to an entire data structure, i.e. an array, a search tree etc.. One can differentiate between three classes of index structures depending on the manner of organization:

1. **Data-organizing index structures** are used to organize the actual amount of data. They mostly rely on *search trees*.
2. **Space-organizing index structures** are used to organize the space that holds the data. They make use of *dynamic hashing*.
3. **Hybrid index structures** are a combination of both classes. They are based on *hash trees*.

There are several requirements for an index structure in order to meet its purpose.

- *Efficient search:* A data query on the index structure should return an answer in optimal time, i.e. the query should be directed to the page or pages that contain the queried data using as little steps as possible.
- *Dynamic insertion, deletion and modification of data sets:* The amount of data to be organized changes over time, leading to alterations in the index structure as well. Any implementation requiring a complete reorganization of the index structure on insertion, deletion or modification of data sets is clearly unacceptable. Any of these operations may therefore only lead to local changes.
- *Local preservation of order:* If there are some data sets the keys of which are successors within the applied order relation (i.e. the less-or-equal relation on non-negative integers), this order should be preserved within the index structure. This holds for search trees but it does not hold for linear hashing. It is clearly of great importance regarding the application scenario in question.

- *Efficient use of space:* This requirement is of great importance for real-world applications. So far the reference implementation *AMBIENCE* has served as a proof of concept. Accordingly the number of messages, i.e. the amount of data to be queried, has been relatively small compared to a real-world scenario. Therefore the memory requirements of any index structure within the current scenario that represents the actual amount of data is unlikely to require vast amounts of memory. However, keeping in mind future application scenarios for *AMBIENCE*, efficient use of space cannot be entirely discarded.

Further requirements include *feasability* and *implementation cost.* Any index structure aiming at a real-world implementation such as *AMBIENCE* naturally has to be feasable, so this requirement will be overlooked in the following. As this work clearly has a scholarly background, not an industrial one, the implementation cost will be disregarded as well. [OW12]

2.2.1 B- und B+-Bäume

[Knu98]

2.3 AMBIENCE

[WDS15].

3 Vewandte Themen

4 Eigene Arbeiten

5 Evaluation

6 Zusammenfassung und Ausblick

Literaturverzeichnis

[ADI⁺12] AHLGREN, BENGT, CHRISTIAN DANNEWITZ, CLAUDIO IMBRENDA, DIRK
KUTSCHER und BÖRJE OHLMAN: *A Survey of Information-Centric Networking.* Communications Magazine, IEEE, 50(7):26–36, 2012.

[AT06] AGARWAL, SACHIN und ARI TRACHTENBERG: *Approximating the number of differences between remote sets.* In: *Information Theory Workshop, 2006. ITW '06 Punta del Este. IEEE*, Seiten 217–221, March 2006.

[BCM02] BYERS, JOHN, JEFFREY CONSIDINE und MICHAEL MITZENMACHER: *Fast Approximate Reconciliation of Set Differences.* In: *BU Computer Science TR*, Seiten 2002–2019, 2002.

[BM04] BRODER, ANDREI und MICHAEL MITZENMACHER: *Network Applications of Bloom Filters: A Survey.* Internet Mathematics, 1(4):485–509, 2004.

[BMS07] BAYARDO, ROBERTO, YIMING MA und RAMAKRISHNAN SRIKANT: *Scaling Up All Pairs Similarity Search.* In: *Proceedings of the 16th international conference on World Wide Web*, Seiten 131–140. ACM, 2007.

[DWM10] DÜRR, MICHAEL, MARTIN WERNER und MARCO MAIER: *Re-Socializing Online Social Networks.* In: *Green Computing and Communications (GreenCom), 2010 IEEE/ACM International Conference on & International Conference on Cyber, Physical and Social Computing (CPSCom)*, Seiten 786–791. IEEE, 2010.

[FCAB00] FAN, LI, PEI CAO, JUSSARA ALMEIDA und ANDREI BRODER: *Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol.* IEEE/ACM Transactions on Networking (TON), 8(3):281–293, 2000.

[HP94] HELLERSTEIN, JOSEPH und AVI PFEFFER: *The RD-Tree: An Index Structure for Sets.* Technischer Bericht, University of Wisconsin-Madison, Computer Sciences Department, 1994.

[Jan95] JANNINK, JAN: *Implementing Deletion in B+-Trees.* ACM Sigmod Record, 24(1):33–38, 1995.

[Knu98] KNUTH, DONALD: *The art of computer programming, Volume 3, Sorting and searching.* Addison Wesley Longman, 1998.

[LC86] LEHMAN, TOBIN und MICHAEL CAREY: *A Study of Index Structures for Main Memory Database Management Systems.* In: *Proc. VLDB*, 1986.

[Mit02] MITZENMACHER, MICHAEL: *Compressed Bloom Filters.* IIEEE/ACM Transactions on Networking (TON), 10(5):604–612, 2002.

[Naf05] NAFE, CLEMENS: *Indexierung lokaler Daten in Peer-to-Peer-Netzwerken.* Diplomarbeit, Universität Rostock, 2005.

Literaturverzeichnis

[OW12]   Ottmann, Thomas und Peter Widmayer: *Algorithmen und Datenstrukturen.* Spektrum Akademischer Verlag, 5 Auflage, 2012.

[QLC14]   Qiao, Yan, Tao Li und Shigang Chen: *Fast Bloom Filters and their Generalization.* Parallel and Distributed Systems, IEEE Transactions on, 25(1):93–103, Januar 2014.

[RK14]   Ruppel, Peter und Axel Küpper: *Geocookie: A Space-Efficient Representation of Geographic Location Sets.* Journal of Information Processing, 22(3):418–424, 2014.

[SBE+12] Sarwat, Mohamed, Jie Bao, Ahmed Eldawy, Justin Levandoski, Amr Magdy und Mohamed Mokbel: *Sindbad: A Location-(B)ased Social Networking System.* In: *Proceedings or the 2012 ACM SIGMOD International Conference on Management of Data*, Seiten 649–652. ACM, 2012.

[Sch13]   Schnell, Rainer: *Getting Big Data But Avoiding Big Brother.* WP-GRLC, 2, 2013.

[SS11]   Sakuma, Hiroshi und Fumiako Sato: *Evaluation of the Structured Bloom Filters Based on Similarity.* In: *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, Seiten 316–323, März 2011.

[STT+09] Shiraki, Toru, Yuichi Teranishi, Susumu Takeuchi, Kaname Harumoto und Shojiro Nishio: *A Bloom Filter-Based User Search Method Based on Movement Records for P2P Network.* In: *Applications and the Internet, 2009. SAINT '09. Ninth International Symposium on*, Seiten 177–180. IEEE, Juli 2009.

[SW14]   Schönfeld, Mirco und Martin Werner: *Node Wake-(U)p via OVSF-Coded Bloom Filters in Wireless Sensor Networks.* In: *Ad Hoc Networks*, Seiten 119–134. Springer, 2014.

[WDS15]   Werner, Martin, Florian Dorfmeister und Mirco Schönfeld: *AMBIENCE: A Context-Centric Online Social Network.* In: *12th IEEE Workshop on Positioning, Navigation and Communications (WPNC '15)*, 2015.

[YL02]   Yang, Congjun und King-Ip Lin: *An Index Structure for Improving Closest Pairs and Related Join Queries in Spatial Databases.* In: *Database Engineering and Applications Symposium, 2002. Proceedings. International*, Seiten 140–149. IEEE, 2002.

[Zha12]   Zhang, Zhenghao: *Analog Bloom Filter: Efficient simultaneous query for wireless networks.* In: *Global Communications Conference (GLOBECOM), 2012 IEEE*, Seiten 3340–3346. IEEE, 2012.

[ZJW04]   Zhu, Yifeng, Hong Jiang und Jun Wang: *Hierarchical Bloom Filter Arrays (HBA): A Novel, Scalable Metadata Management System for Large Cluster-based Storage.* In: *Cluster Computing, 2004 IEEE International Conference on*, Seiten 165–174. IEEE, 2004.