LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR MOBILE UND VERTEILTE SYSTEME

MASTERARBEIT

Titel der Arbeit

Judith Greif

Entwurf vom 25. November 2015

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
INSTITUT FÜR INFORMATIK
LEHRSTUHL FÜR MOBILE UND VERTEILTE SYSTEME

MASTERARBEIT

Titel der Arbeit

Judith Greif

Aufgabenstellerin:	Prof. Dr. Claudia Linnhoff-Popien
Betreuer:	Mirco Schönfeld Dr. Martin Werner
Abgabetermin:	1. Januar 2099

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

München, den 1. Januar 2099

. . . . . .
(Unterschrift des Kandidaten)

Abstract

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

Inhaltsverzeichnis

1 Introduction

References up to now: [AT06], [ADI+12], [BMS07], [BM04], [BCM02], [DWM10], [HP94], [LC86], [Naf05], [QLC14], [RK14], [SBE+12], [Sch13], [SW14], [STT+09], [WDS15], [YL02], [Zha12], [ZJW04].

1.1 Problem Statement

2 Background

2.1 Bloom Filters

2.1.1 Classic Bloom Filter

[Blo70]

2.1.2 Bloom Filter Operations and Variants

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

2.2 Mathematic Principles

2.3 Index Structures in Database Systems

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]

2.3.1 B-Tree

[Knu98]

2.3.2 R-Tree

2.3.3 R*-Tree

2.3.4 Heap

2.4 AMBIENCE

[WDS15].

3 Implementation

4 Evaluation

4.1 Test Data

5 Conclusions

Literaturverzeichnis

[ADI+12] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Börje Ohlman. A survey of information-centric networking. *Communications Magazine, IEEE*, 50(7):26–36, 2012.

[AT06] S. Agarwal and A. Trachtenberg. Approximating the number of differences between remote sets. In *Information Theory Workshop, 2006. ITW '06 Punta del Este. IEEE*, pages 217–221, March 2006.

[BCM02] John Byers, Jeffrey Considine, and Michael Mitzenmacher. Fast Approximate Reconciliation of Set Differences. In *BU Computer Science TR*, pages 2002–2019, 2002.

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[BM04] Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics*, 1(4):485–509, 2004.

[BMS07] Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. Scaling up all pairs similarity search. In *Proceedings of the 16th international conference on World Wide Web*, pages 131–140. ACM, 2007.

[DWM10] Michael Dürr, Martin Werner, and 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)*, pages 786–791. IEEE, 2010.

[FCAB00] Li Fan, Pei Cao, Jussara Almeida, and Andrei Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking (TON)*, 8(3):281–293, 2000.

[HP94] Joseph M. Hellerstein and Avi Pfeffer. The RD-Tree: An Index Structure for Sets. Technical report, University of Wisconsin-Madison, Computer Sciences Department, 1994.

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

[LC86] Tobin J. Lehman and Michael J. Carey. A study of index structures for main memory database management systems. In *Proc. VLDB*, 1986.

[Mit02] Michael Mitzenmacher. Compressed bloom filters. *IIEEE/ACM Transactions on Networking (TON)*, 10(5):604–612, 2002.

[Naf05] Clemens Nafe. Indexierung lokaler Daten in Peer-to-Peer-Netzwerken. Master's thesis, Universität Rostock, 2005.

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

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

[RK14] Peter Ruppel and Axel Küpper. Geocookie: a space-efficient representation of geographic location sets. *Journal of Information Processing*, 22(3):418–424, 2014.

[SBE+12] Mohamed Sarwat, Jie Bao, Ahmed Eldawy, Justin J Levandoski, Amr Magdy, and Mohamed F Mokbel. Sindbad: a location-based social networking system. In *Proceedings or the 2012 ACM SIGMOD International Conference on Management of Data*, pages 649–652. ACM, 2012.

[Sch13] Rainer Schnell. Getting Big Data but avoiding Big Brother. *WP-GRLC*, 2, 2013.

[SS11] H. Sakuma and F. Sato. Evaluation of the Structured Bloom Filters Based on Similarity. In *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, pages 316–323, March 2011.

[STT+09] Toru Shiraki, Yuichi Teranishi, Susumu Takeuchi, Kaname Harumoto, and 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*, pages 177–180. IEEE, July 2009.

[SW14] Mirco Schönfeld and Martin Werner. Node wake-up via ovsf-coded bloom filters in wireless sensor networks. In *Ad Hoc Networks*, pages 119–134. Springer, 2014.

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

[YL02] Congjun Yang and 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*, pages 140–149. IEEE, 2002.

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

[ZJW04] Yifeng Zhu, Hong Jiang, and 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*, pages 165–174. IEEE, 2004.