

From Big Data Theory to Big Data Practice

Martin Farach-Colton^{*1}, Fabian Daniel Kuhn^{*2}, Ronitt Rubinfeld^{*3},
and Przemysław Uznański^{*4}

1 Rutgers University – Piscataway, US. martin@farach-colton.com

2 Universität Freiburg, DE. kuhn@cs.uni-freiburg.de

3 MIT – Cambridge, US. ronitt@csail.mit.edu

4 Pathway – Wrocław, PL. przemek@pathway.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23071 “From Big Data Theory to Big Data Practice”. Some recent advances in the theory of algorithms for big data – sublinear/local algorithms, streaming algorithms and external memory algorithms – have translated into impressive improvements in practice, whereas others have remained stubbornly resistant to useful implementations. This seminar aimed to glean lessons for those aspect of these algorithms that have led to practical implementation to see if the lessons learned can both improve the implementations of other theoretical ideas and to help guide the next generation of theoretical advances.

Seminar February 12–17, 2023 – <https://www.dagstuhl.de/23071>

2012 ACM Subject Classification Information systems → Data structures; Theory of computation → Data structures design and analysis; Theory of computation → Distributed algorithms; Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases external memory, local algorithms, sublinear algorithms

Digital Object Identifier 10.4230/DagRep.13.2.33

1 Executive Summary

Martin Farach-Colton

Fabian Daniel Kuhn

Ronitt Rubinfeld

Przemysław Uznański

License  Creative Commons BY 4.0 International license

© Martin Farach-Colton, Fabian Daniel Kuhn, Ronitt Rubinfeld, and Przemysław Uznański

As data has grown faster than RAM, the theory of algorithms has expanded to provide approaches for tackling such problems. These fall into three broad categories:

- Streaming and semi-streaming algorithms
- Sublinear or local algorithms
- External memory algorithms

Each of these areas has a vibrant literature, and many of the results from the theory literature have made their way into practice. Other results are not suitable for implementation and deployment.

This Dagstuhl Seminar’s aim was to address several questions by bringing together algorithmicists from these subcommunities, as well as algorithms engineers. Specifically, the aim was to address the following questions:

* Editor / Organizer



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 4.0 International license

From Big Data Theory to Big Data Practice, *Dagstuhl Reports*, Vol. 13, Issue 2, pp. 33–46

Editors: Martin Farach-Colton, Fabian Daniel Kuhn, Ronitt Rubinfeld, and Przemysław Uznański



DAGSTUHL
REPORTS

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- What themes emerge from considering practical algorithms from the theory literature?
- Can we use these insights to create new models or to capture interesting new optimization criteria?

By bringing together researchers in these disparate areas and by including researchers in algorithms engineering, we hope to have brought to light these deep connections. The goals were to:

- Extract shared lessons to help guide theoretical research towards practical solutions;
- Create a feedback loop where commonalities of practical solutions can help guide future theoretical research;
- Help cross-pollinate these research areas.

Organization of the Seminar

The seminar brought together 38 researchers from theoretical computer science and systems research, both from academia and from industry. The participants consisted of both senior and junior researchers, including a number of postdocs and advanced PhD students.

During the four¹ days of the seminar, 20 talks of different lengths took place. Speakers were given no time constraints, but all the talks took between 30 to 60 minutes. We value the freedom given to the speakers, and the audience enjoyed the relaxed schedule as well. Each day afternoons were spent on fruitful discussions and open-ended open problems sessions.

Outcome

The seminar was seen as a success by organizers and participants. It brought together relevant communities, shared state-of-the-art research, and discussed challenges. The talks were of high quality and stimulating, encouraging participants to actively engage in working groups during the afternoon and evenings. It was particularly appreciated that a significant number of younger researchers (postdocs and PhD students) participated and integrated well. The organizers would like to express their gratitude to the Scientific Directorate and the administration of the Dagstuhl Center for their strong support during the seminar.

¹ The seminar had to be cut short by one day due to unforeseen logistic issues.

2 Table of Contents

Executive Summary

<i>Martin Farach-Colton, Fabian Daniel Kuhn, Ronitt Rubinfeld, and Przemysław Uznański</i>	33
--	----

Overview of Talks

Paging on Shared Caches <i>Kunal Agrawal</i>	37
Contention resolution without collision detection <i>Michael A. Bender</i>	37
Data Structures on the Ultra-Wide Word RAM <i>Philip Bille</i>	37
Mosaic Pages <i>Alexander Conway</i>	38
Sampling Edges in Sublinear Time <i>Talya Eden</i>	38
Tiny Pointers <i>Martin Farach-Colton</i>	39
Atomic Power in Forks: A Super-Logarithmic Lower Bound for Implementing Butterfly Networks in the Nonatomic Binary Fork-Join Model <i>Riko Jacob</i>	39
Online List Labeling: Breaking the $\log^2(n)$ Barrier <i>Hanna Koplós</i>	40
Linear Probing Revisited: The Demise of Clustering <i>William Kuszmaul</i>	40
Cardinality Estimation Using Gumbel Distribution <i>Aleksander Łukasiewicz</i>	41
Big data algorithms for distance computation <i>Yasamin Nazari</i>	41
Locality and Volume in the Context of Algorithms for Dynamic Data Sets <i>Krzysztof Nowicki</i>	42
Adaptive Adversaries vs. Randomized Algorithms: A Few Challenges <i>Krzysztof Onak</i>	42
Massively-Parallel Computing in a Heterogenous Regime <i>Rotem Oshman</i>	42
IcebergHT: High Performance PMEM Hash Tables Through Stability and Low Associativity Authors: Prashant Pandey <i>Prashant Pandey</i>	43
Can sublinear graph algorithms impact practice? <i>Comandur Seshadhri</i>	44
Locality in online, dynamic, sequential, and distributed graph algorithms <i>Jukka Suomela</i>	44

Dynamic Graph Sketching: To Infinity And Beyond <i>David Tench</i>	44
Reactive incremental algorithms with Pathway. <i>Przemysław Uznański</i>	45
Participants	46

3 Overview of Talks

3.1 Paging on Shared Caches

Kunal Agrawal (Washington University – St. Louis, US)

License © Creative Commons BY 4.0 International license
© Kunal Agrawal

The talk presents results on using shared caches where p disjoint request sequences are trying to access a cache concurrently.

3.2 Contention resolution without collision detection

Michael A. Bender (Stony Brook University, US)

License © Creative Commons BY 4.0 International license
© Michael A. Bender

Joint work of Michael A. Bender, Tsvi Kopelowitz, William Kuszmaul, Seth Pettie

Main reference Michael A. Bender, Tsvi Kopelowitz, William Kuszmaul, Seth Pettie: “Contention resolution without collision detection”, in Proc. of the Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020, pp. 105–118, ACM, 2020.

URL <https://doi.org/10.1145/3357713.3384305>

In this talk we revisit the classical problem of randomized exponential backoff on a multiple-access channel. We show how to achieve constant throughput even when the players cannot detect collisions, i.e, simultaneous transmissions, on the channel.

3.3 Data Structures on the Ultra-Wide Word RAM

Philip Bille (Technical University of Denmark – Lyngby, DK)

License © Creative Commons BY 4.0 International license
© Philip Bille

Joint work of Philip Bille, Inge Li Gørtz, Tord Stordalen

Main reference Philip Bille, Inge Li Gørtz, Tord Stordalen: “Predecessor on the Ultra-Wide Word RAM”, in Proc. of the 18th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2022, June 27-29, 2022, Tórshavn, Faroe Islands, LIPIcs, Vol. 227, pp. 18:1–18:15, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

URL <https://doi.org/10.4230/LIPIcs.SWAT.2022.18>

We consider the predecessor problem on the ultra-wide word RAM model of computation, which extends the word RAM model with ultrawords consisting of w^2 bits [TAMC, 2015]. The model supports arithmetic and boolean operations on ultrawords, in addition to scattered memory operations that access or modify w (potentially non-contiguous) memory addresses simultaneously. The ultra-wide word RAM model captures (and idealizes) modern vector processor architectures. Our main result is a simple, linear space data structure that supports predecessor in constant time and updates in amortized, expected constant time. This improves the space of the previous constant time solution that uses space in the order of the size of the universe. Our result holds even in a weaker model where ultrawords consist of $w^{(1+\varepsilon)}$ bits for any $\varepsilon > 0$. It is based on a new implementation of the classic x-fast trie data structure of Willard [Inform. Process. Lett. 17(2), 1983] combined with a new dictionary data structure that supports fast parallel lookups.

3.4 Mosaic Pages

Alexander Conway (VMware Research – Palo Alto, US)

License © Creative Commons BY 4.0 International license
© Alexander Conway

Joint work of Krishnan Gosakan, Jaehyun Han, William Kuszmaul, Ibrahim N. Mubarek, Nirjhar Mukherjee, Karthik Sriram, Guido Tagliavini, Evan West, Michael A. Bender, Abhishek Bhattacharjee, Alex Conway, Martin Farach-Colton, Jayneel Gandhi, Rob Johnson, Sudarsun Kannan, Donald E. Porter

Main reference Krishnan Gosakan, Jaehyun Han, William Kuszmaul, Ibrahim N. Mubarek, Nirjhar Mukherjee, Karthik Sriram, Guido Tagliavini, Evan West, Michael A. Bender, Abhishek Bhattacharjee, Alex Conway, Martin Farach-Colton, Jayneel Gandhi, Rob Johnson, Sudarsun Kannan, Donald E. Porter: “Mosaic Pages: Big TLB Reach with Small Pages”, in Proc. of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, ASPLOS 2023, Vancouver, BC, Canada, March 25-29, 2023, pp. 433–448, ACM, 2023.

URL <https://doi.org/10.1145/3582016.3582021>

In this talk, I present an algorithmic approach to co-designing TLB hardware and the paging mechanism to increase TLB reach without the fragmentation issues incurred by huge pages. Along the way, I’ll introduce a new hash-table design that overcomes existing tradeoffs, and achieves better performance than state-of-the-art hash tables both in theory and in practice. Key to these results are “tiny pointers,” an algorithmic technique for compressing pointers.

3.5 Sampling Edges in Sublinear Time

Talya Eden (Bar-Ilan University – Ramat Gan, IL)

License © Creative Commons BY 4.0 International license
© Talya Eden

Joint work of Talya Eden, Shyam Narayanan, Jakub Tetek

Main reference Talya Eden, Shyam Narayanan, Jakub Tetek: “Sampling an Edge in Sublinear Time Exactly and Optimally”, in Proc. of the 2023 Symposium on Simplicity in Algorithms, SOSA 2023, Florence, Italy, January 23-25, 2023, pp. 253–260, SIAM, 2023.

URL <https://doi.org/10.1137/1.9781611977585.ch23>

Sampling edges from a graph in sublinear time is a fundamental problem and a powerful subroutine for designing sublinear-time algorithms. Suppose we have access to the vertices of the graph and know a constant-factor approximation to the number of edges. An algorithm for pointwise ε -approximate edge sampling with complexity $O(n/\sqrt{\varepsilon m})$ has been given by Eden and Rosenbaum [SOSA 2018]. This has been later improved by Tětek and Thorup [STOC 2022] to $O(n \log(\varepsilon^{-1})/\sqrt{m})$. At the same time, $\Omega(n/\sqrt{m})$ time is necessary. We close the problem, by giving an algorithm with complexity $O(n/\sqrt{m})$ for the task of sampling an edge exactly uniformly.

3.6 Tiny Pointers

Martin Farach-Colton (Rutgers University – Piscataway, US)

License © Creative Commons BY 4.0 International license
© Martin Farach-Colton

Joint work of Michael A. Bender, Alex Conway, Martin Farach-Colton, William Kuszmaul, Guido Tagliavini

Main reference Michael A. Bender, Alex Conway, Martin Farach-Colton, William Kuszmaul, Guido Tagliavini: “Tiny Pointers”, in Proc. of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023, pp. 477–508, SIAM, 2023.

URL <https://doi.org/10.1137/1.9781611977554.ch21>

We introduce a new data-structural object that we call the tiny pointer. In many applications, traditional $\log n$ -bit pointers can be replaced with $o(\log n)$ -bit tiny pointers at the cost of only a constant-factor time overhead and a small probability of failure. We develop a comprehensive theory of tiny pointers, and give optimal constructions for both fixed-size tiny pointers (i.e., settings in which all of the tiny pointers must be the same size) and variable-size tiny pointers (i.e., settings in which the average tiny-pointer size must be small, but some tiny pointers can be larger). If a tiny pointer references an element in an array filled to load factor $1 - \delta$, then the optimal tiny-pointer size is $\Theta(\log \log \log n + \log \delta^{-1})$ bits in the fixed-size case, and $\Theta(\log \delta^{-1})$ expected bits in the variable-size case. Using tiny pointers, it is possible to improve succinctness bounds on a variety of classic data-structure problems.

3.7 Atomic Power in Forks: A Super-Logarithmic Lower Bound for Implementing Butterfly Networks in the Nonatomic Binary Fork-Join Model

Riko Jacob (IT University of Copenhagen, DK)

License © Creative Commons BY 4.0 International license
© Riko Jacob

Joint work of Michael T. Goodrich, Riko Jacob, Nodari Sitchinava

Main reference Michael T. Goodrich, Riko Jacob, Nodari Sitchinava: “Atomic Power in Forks: A Super-Logarithmic Lower Bound for Implementing Butterfly Networks in the Nonatomic Binary Fork-Join Model”, in Proc. of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 – 13, 2021, pp. 2141–2153, SIAM, 2021.

URL <https://doi.org/10.1137/1.9781611976465.128>

We prove an $\Omega(\log n \log \log n)$ lower bound for the span of implementing the n input, $\log n$ -depth FFT circuit (also known as butterfly network) in the nonatomic binary fork-join model. In this model, memory-access synchronizations occur only through fork operations, which spawn two child threads, and join operations, which resume a parent thread when its child threads terminate. Our bound is asymptotically tight for the nonatomic binary fork-join model, which has been of interest of late, due to its conceptual elegance and ability to capture asynchrony. Our bound implies super-logarithmic lower bound in the nonatomic binary fork-join model for implementing the butterfly merging networks used, e.g., in Batcher’s bitonic and odd-even mergesort networks. This lower bound also implies an asymptotic separation result for the atomic and nonatomic versions of the fork-join model, since, as we point out, FFT circuits can be implemented in the atomic binary fork-join model with span equal to their circuit depth.

3.8 Online List Labeling: Breaking the $\log^2(n)$ Barrier

Hanna Komlós (Rutgers University – New Brunswick, US)

License  Creative Commons BY 4.0 International license
 Hanna Komlós

Joint work of Michael A. Bender, Alex Conway, Martin Farach-Colton, Hanna Komlós, William Kuszmaul, Nicole Wein

Main reference Michael A. Bender, Alex Conway, Martin Farach-Colton, Hanna Komlós, William Kuszmaul, Nicole Wein: “Online List Labeling: Breaking the $\log^2 n$ Barrier”, in Proc. of the 63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 – November 3, 2022, pp. 980–990, IEEE, 2022.

URL <https://doi.org/10.1109/FOCS54457.2022.00096>

The online list-labeling problem is a classical combinatorial problem with a large literature of upper bounds, lower bounds, and applications. The goal is to store a dynamically-changing set of n items in an array of m slots, while maintaining the invariant that the items appear in sorted order, and while minimizing the relabeling cost, defined to be the number of items that are moved per insertion/deletion. There has long existed a gap between the lower and upper bounds in the most algorithmically interesting part of the problem’s parameter space. We present our recent results, which narrow this gap for the first time in nearly 4 decades.

3.9 Linear Probing Revisited: The Demise of Clustering

William Kuszmaul (MIT – Cambridge, US)

License  Creative Commons BY 4.0 International license
 William Kuszmaul

Joint work of Michael A. Bender, Bradley C. Kuszmaul, William Kuszmaul

Main reference Michael A. Bender, Bradley C. Kuszmaul, William Kuszmaul: “Linear Probing Revisited: Tombstones Mark the Demise of Primary Clustering”, in Proc. of the 62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022, pp. 1171–1182, IEEE, 2021.

URL <https://doi.org/10.1109/FOCS52979.2021.00115>

The linear-probing hash table is one of the oldest and most widely used data structures in computer science. However, linear probing also famously comes with a major drawback: as soon as the hash table reaches a high memory utilization, elements within the hash table begin to cluster together, causing insertions to become slow. This clustering phenomenon, which was first discovered by Donald Knuth in 1962, increases the expected time per insertion to $\Theta(x^2)$ (rather than the more desirable $\Theta(x)$) in a hash table that is a $1 - 1/x$ fraction full.

A natural question is whether one can somehow reduce clustering. In this paper, we establish an even stronger statement: the classical linear-probing hash table (even as it was first implemented in the 1950s) already has less clustering than the classical results would seem to suggest. As insertions and deletions are performed over time, the tombstones left behind by deletions cause the combinatorial structure of the hash table to stabilize in a way that eliminates clustering. This means that, for some versions of linear probing, the amortized expected time per operation is actually $\tilde{O}(x)$. We also present a new version of linear probing that avoids clustering entirely, achieving $O(x)$ expected time per operation.

3.10 Cardinality Estimation Using Gumbel Distribution

Aleksander Łukasiewicz (University of Wrocław, PL)

License © Creative Commons BY 4.0 International license
© Aleksander Łukasiewicz

Joint work of Aleksander Łukasiewicz, Przemysław Uznański

Main reference Aleksander Łukasiewicz, Przemysław Uznanski: “Cardinality Estimation Using Gumbel Distribution”, in Proc. of the 30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany, LIPIcs, Vol. 244, pp. 76:1–76:13, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

URL <https://doi.org/10.4230/LIPIcs.ESA.2022.76>

Cardinality estimation is the task of approximating the number of distinct elements in a large dataset with possibly repeating elements. LogLog and HyperLogLog (c.f. Durand and Flajolet [ESA 2003], Flajolet et al. [Discrete Math Theor. 2007]) are small space sketching schemes for cardinality estimation, which have both strong theoretical guarantees of performance and are highly effective in practice. This makes them a highly popular solution with many implementations in big-data systems (e.g. Algebird, Apache DataSketches, BigQuery, Presto and Redis). However, despite having simple and elegant formulation, both the analysis of LogLog and HyperLogLog are extremely involved – spanning over tens of pages of analytic combinatorics and complex function analysis. We propose a modification to both LogLog and HyperLogLog that replaces discrete geometric distribution with the continuous Gumbel distribution. This leads to a very short, simple and elementary analysis of estimation guarantees, and smoother behavior of the estimator.

3.11 Big data algorithms for distance computation

Yasamin Nazari (Universität Salzburg, AT)

License © Creative Commons BY 4.0 International license
© Yasamin Nazari

Joint work of Greg Bodwin, Michael Dinitz, Jakub Łącki, Yasamin Nazari

Main reference Yasamin Nazari: “Distributed, Parallel and Dynamic Distance Structures”, Dissertation, Baltimore, 2021.

URL <https://jscholarship.library.jhu.edu/bitstream/handle/1774.2/64360/NAZARI-DISSERTATION-2021.pdf>

In recent years, there has been a growing interest in designing algorithms in models that capture different aspects of modern computational systems and big data processing. This talk focuses on graph algorithms in several such models, namely: distributed models, dynamic models, fault tolerant settings and the massively parallel computation model (MPC). Specifically, we focus on graph distance computation in these models. We study several graph theoretic structures that preserve approximate distances, but tradeoff this approximation factor with size, query time, or the number of hops on the approximate shortest paths. We then show how these combinatorial structures can be utilized for faster distance computation in distributed or dynamic settings. We conclude with applications in other related problems such as clustering and routing.

3.12 Locality and Volume in the Context of Algorithms for Dynamic Data Sets

Krzysztof Nowicki (Pathway – Paris, FR)

License © Creative Commons BY 4.0 International license
© Krzysztof Nowicki

Joint work of Aleksander B. G. Christiansen, Krzysztof D. Nowicki, Eva Rotenberg

Main reference Aleksander B. G. Christiansen, Krzysztof D. Nowicki, Eva Rotenberg: “Improved Dynamic Colouring of Sparse Graphs”, CoRR, Vol. abs/2211.06858, 2022.

URL <https://doi.org/10.48550/arXiv.2211.06858>

In this talk I’ll explain how to make algorithms for dynamic data sets out of distributed algorithms. The talk covers recent improvements on theory of algorithms for big data sets [Improved Dynamic Colouring of Sparse Graphs Aleksander B. G. Christiansen, Krzysztof D. Nowicki, Eva Rotenberg] and addresses some possible practical applications of presented approach [at Pathway.com].

3.13 Adaptive Adversaries vs. Randomized Algorithms: A Few Challenges

Krzysztof Onak (Boston University, US)

License © Creative Commons BY 4.0 International license
© Krzysztof Onak

Joint work of Krzysztof Onak, Omri Ben-Eliezer, Talya Eden, Rajesh Jayaram, Jakub Łącki, Slobodan Mitrović, Piotr Sankowski

Streaming algorithms and data structures often leverage randomness to lower their utilization of resources such as memory and computation. I will talk about the possible danger coming from the fact that estimates or replies provided by a randomized algorithm can leak information about its internal randomness. This information could be exploited via adaptive updates to break the guarantees of the algorithm that hold for any fixed sequences of updates with high probability.

Is there a way to make these types of randomized algorithms perform well even in the presence of adaptive updates? In this talk I will share both positive and negative news, ranging from simple techniques for moment estimation in the streaming model to a generic framework that shows that explicitly maintaining a collection of samples from nearly uniform distribution cannot be achieved with a small number of edits. This is a relatively new direction of research and many fundamental questions remain open.

3.14 Massively-Parallel Computing in a Heterogeneous Regime

Rotem Oshman (Tel Aviv University, IL)

License © Creative Commons BY 4.0 International license
© Rotem Oshman

Joint work of Orr Fischer, Adi Horowitz, Rotem Oshman

Main reference Orr Fischer, Adi Horowitz, Rotem Oshman: “Massively Parallel Computation in a Heterogeneous Regime”, CoRR, Vol. abs/2302.14692, 2023.

URL <https://doi.org/10.48550/arXiv.2302.14692>

Massively-parallel graph algorithms have received extensive attention over the past decade, with research focusing on three memory regimes: the superlinear regime, the near-linear regime, and the sublinear regime. The sublinear regime is the most desirable in practice,

but conditional hardness results point towards its limitations. In this work we study a *heterogeneous* model, where the memory of the machines varies in size. We focus mostly on the heterogeneous setting created by adding a single near-linear machine to the sublinear MPC regime, and show that even a single large machine suffices to circumvent most of the conditional hardness results for the sublinear regime: for graphs with n vertices and m edges, we give (a) an MST algorithm that runs in $O(\log \log(m/n))$ rounds; (b) an algorithm that constructs an $O(k)$ -spanner of size $O(n^{1+1/k})$ in $O(1)$ rounds; and (c) a maximal-matching algorithm that runs in $O(\sqrt{\log(m/n)} \log \log(m/n))$ rounds. We also observe that the best known near-linear MPC algorithms for several other graph problems which are conjectured to be hard in the sublinear regime (minimum cut, maximal independent set, and vertex coloring) can easily be transformed to work in the heterogeneous MPC model with a single near-linear machine, while retaining their original round complexity in the near-linear regime. If the large machine is allowed to have *superlinear* memory, all of the problems above can be solved in $O(1)$ rounds.

3.15 IcebergHT: High Performance PMEM Hash Tables Through Stability and Low Associativity Authors: Prashant Pandey

Prashant Pandey (University of Utah – Salt Lake City, US)

License © Creative Commons BY 4.0 International license
© Prashant Pandey

Joint work of Prashant Pandey, Michael A. Bender, Alex Conway, Martin Farach-Colton, William Kuszmaul, Guido Tagliavini, Rob Johnson

Main reference Prashant Pandey, Michael A. Bender, Alex Conway, Martin Farach-Colton, William Kuszmaul, Guido Tagliavini, Rob Johnson: “IcebergHT: High Performance PMEM Hash Tables Through Stability and Low Associativity”, CoRR, Vol. abs/2210.04068, 2022.

URL <https://doi.org/10.48550/arXiv.2210.04068>

Modern hash table designs strive to minimize space while maximizing speed. The most important factor in speed is the number of cache lines accessed during updates and queries. This is especially important on PMEM, which is slower than DRAM and in which writes are more expensive than reads. This paper proposes two stronger design objectives: stability and low-associativity. A stable hash table doesn’t move items around, and a hash table has low associativity if there are only a few locations where an item can be stored. Low associativity ensures that queries need to examine only a few memory locations, and stability ensures that insertions write to very few cache lines. Stability also simplifies scaling and crash safety. We present IcebergHT, a fast, crash-safe, concurrent, and space-efficient hash table for PMEM based on the design principles of stability and low associativity. IcebergHT combines in-memory metadata with a new hashing technique, iceberg hashing, that is (1) space efficient, (2) stable, and (3) supports low associativity. In contrast, existing hash-tables either modify numerous cache lines during insertions (e.g. cuckoo hashing), access numerous cache lines during queries (e.g. linear probing), or waste space (e.g. chaining). Moreover, the combination of (1)-(3) yields several emergent benefits: IcebergHT scales better than other hash tables, supports crash-safety, and has excellent performance on PMEM (where writes are particularly expensive). In our benchmarks, IcebergHT inserts are 50% to $3\times$ faster than state-of-the-art PMEM hash tables Dash and CLHT and queries are 20% to $2\times$ faster. IcebergHT space overhead is 17%, whereas Dash and CLHT have space overheads of $2\times$ and $3\times$, respectively. IcebergHT also exhibits linear scaling and is crash safe. In DRAM, IcebergHT outperforms state-of-the-art hash tables libcuckoo and CLHT by almost $2\times$ on insertions while offering good query throughput and much better space efficiency.

3.16 Can sublinear graph algorithms impact practice?

Comandur Seshadhri (University of California – Santa Cruz, US)

License  Creative Commons BY 4.0 International license
 Comandur Seshadhri

This talk is about sublinear graph algorithms applied to the analysis of real-world graphs. We discuss a provable sublinear algorithms for estimating the degree distribution in sublinear time. To analyze this algorithm, we define fatness indices of the degree distribution. The running time of the algorithm is parameterized by these indices. The algorithm is also practical.

We discuss the overall challenge of designing practical sublinear graph algorithms.

3.17 Locality in online, dynamic, sequential, and distributed graph algorithms

Jukka Suomela (Aalto University, FI)

License  Creative Commons BY 4.0 International license
 Jukka Suomela

Joint work of Amirreza Akbari, Henrik Lievonen, Darya Melnyk, Joona Sarkijarvi, Jukka Suomela
Main reference Amirreza Akbari, Henrik Lievonen, Darya Melnyk, Joona Sarkijarvi, Jukka Suomela: “Online Algorithms with Lookaround”, CoRR, Vol. abs/2109.06593, 2021.
URL <https://arxiv.org/abs/2109.06593>

I will discuss the notion of locality in the context of graph problems, from four different perspectives: online graph algorithms, dynamic graph algorithms, sequential distributed algorithms, and parallel distributed algorithms. I will use the graph coloring problem as a running example, and I will explore settings like this:

- Online graph algorithms: The adversary reveals the graph one node at a time. How far do you need to see around a node until it is safe to pick its color?
- Dynamic graph algorithms: The adversary constructs the graph one edge at a time, and you need to maintain a valid coloring after each addition. How far around the new edge do you need to modify the solution?
- Distributed graph algorithms: Each node has to choose its own color simultaneously in parallel based on its local neighborhood. How far does it need to see?

While these are different questions in general, we can show that there are families of graph problems for which all these notions are equal to each other.

3.18 Dynamic Graph Sketching: To Infinity And Beyond

David Tench (Rutgers University – Piscataway, US)

License  Creative Commons BY 4.0 International license
 David Tench

Existing graph stream processing systems must store the graph explicitly in RAM which limits the scale of graphs they can process. The graph semi-streaming literature offers algorithms which avoid this limitation via linear sketching data structures that use small (sublinear) space, but these algorithms have not seen use in practice to date. In this talk I

will explore what is needed to make graph sketching algorithms practically useful, and as a case study present a sketching algorithm for connected components and a corresponding high-performance implementation. Finally, I will give an overview of the many open problems in this area, focusing on improving query performance of graph sketching algorithms.

3.19 Reactive incremental algorithms with Pathway.

Przemysław Uznański (Pathway – Wrocław, PL)

License © Creative Commons BY 4.0 International license
© Przemysław Uznański

In this talk I discuss the ideas of incremental data processing, reactive algorithms, and their connections to dynamic algorithms, distributed algorithms, and big data processing.

Participants

- Kunal Agrawal
Washington University –
St. Louis, US
- Alkida Balliu
Gran Sasso Science Institute –
L'Aquila, IT
- Michael A. Bender
Stony Brook University, US
- Philip Bille
Technical University of Denmark
– Lyngby, DK
- Sebastian Brandt
CISPA – Saarbrücken, DE
- Vincent Cohen-Addad
Google – Paris, FR
- Alexander Conway
VMware Research –
Palo Alto, US
- Martin Dietzfelbinger
TU Ilmenau, DE
- Anne Driemel
Universität Bonn, DE
- Talya Eden
Bar-Ilan University –
Ramat Gan, IL
- Martin Farach-Colton
Rutgers University –
Piscataway, US
- Inge Li Gørtz
Technical University of Denmark
– Lyngby, DK
- Riko Jacob
IT University of
Copenhagen, DK
- Rob Johnson
VMware – Palo Alto, US
- Hanna Komlós
Rutgers University – New
Brunswick, US
- Adrian Kosowski
Pathway – Paris, FR
- Fabian Daniel Kuhn
Universität Freiburg, DE
- William Kuszmaul
MIT – Cambridge, US
- Aleksander Lukaszewicz
University of Wroclaw, PL
- Petra Mutzel
Universität Bonn, DE
- Yasamin Nazari
Universität Salzburg, AT
- Krzysztof Nowicki
Pathway – Paris, FR
- Dennis Olivetti
Gran Sasso Science Institute –
L'Aquila, IT
- Krzysztof Onak
Boston University, US
- Rotem Oshman
Tel Aviv University, IL
- Prashant Pandey
University of Utah –
Salt Lake City, US
- Ely Porat
Bar-Ilan University –
Ramat Gan, IL
- Simon J. Puglisi
University of Helsinki, FI
- Sofya Raskhodnikova
Boston University, US
- Eva Rotenberg
Technical University of Denmark
– Lyngby, DK
- Ronitt Rubinfeld
MIT – Cambridge, US
- Christian Scheideler
Universität Paderborn, DE
- Comandur Seshadhri
University of California –
Santa Cruz, US
- Christian Sohler
Universität Köln, DE
- Tatiana Starikovskaya
ENS – Paris, FR
- Jukka Suomela
Aalto University, FI
- David Tench
Rutgers University –
Piscataway, US
- Przemyslaw Uznanski
University of Wroclaw, PL

