Trevor A. Brown, Associate Professor at the University of Waterloo

Cheriton School of Computer Science
200 University Ave W
Waterloo, ON N2L 3G1, Canada

E-mail: me [at] tbrown [dot] pro
Personal site: http://tbrown.pro
Lab site: http://mc.uwaterloo.ca

Research Interests

- Concurrent data structures
- Non-blocking algorithms
- Memory management
- Non-volatile memory
- Transactional memory

Employment

Associate Professor. University of Waterloo (Jul 2024 – present).
Affiliated with Algorithms and Complexity, Systems and Networking, and Programming Languages. Heading the Multicore Lab.

Assistant Professor. University of Waterloo (Sep 2018 – Jun 2024).

Postdoctoral Researcher. IST Austria (Oct 2017 – Aug 2018).
Working with Professor Dan Alistarh.

Postdoctoral Researcher. Technion, Israel Institute of Technology (Mar – Oct 2017).
Worked with Professor Hagit Attiya. (A short reciprocal visit to work with Maya Arbel.)

Research Intern. Oracle Labs East, Scalable Synchronization Group (Summer 2015).
Designed and implemented adaptive transactional lock-elision algorithms and work delegation algorithms for large scale systems with non-uniform memory architectures.

Application Developer. UPS Canada (2005 – 2006, 2008 – 2009).
Scaled software to a 15x larger user base; contributed ~200,000 lines of production code.

Education

University of Toronto, Toronto, Canada, Sep. 2011 – Jun. 2017
PhD & MSc Computer Science (supervised by Faith Ellen)
PhD thesis: Techniques for constructing efficient lock-free data structures (312 pages)
MSc thesis: Pragmatic primitives for non-blocking data structures

York University, First Class with Distinction, Toronto, Canada, 2006 – 2011
B.Sc. Hons. Major CS, Hons. Minor Math
Faculty Silver Medal; Ruth Hill Memorial Award (top student in the faculty)

Recognition

- **Best paper candidate** (top 3) at SPAA (2022).
- **Best artifact** at PPoPP (2022).
- **Best artifact** at PPoPP (2021).
- **Best paper candidate** (top 4) at PPoPP (2021).
- **Best paper** at PPoPP (2020).
- Approaching $1M in research funding.
- *Nominated* for Governor General's Gold Medal (U of T, 2017)
  (I did not win, but a school can only nominate one CS student annually)
- Award for Excellence in Teaching Assistance (U of T CS Student Union, 2014).
- Scholarships worth $200,000+.

Summary of Contributions

	In prep / submitted	Published / completed	
		Since 2018	**All time**
Conference paper	4	**18***	28‡
Journal paper		2	2
Short paper / poster		2	9‡
Full manuscript / tech report		7	17
Book		1	1
Patent		1	1
Total	**4**	**31**	**58**
Conference / workshop talks		4	18
Invited talks		2†	3†
Guest lectures		1	3
Seminars / other		7	15
Total		**13**	**39**

- * includes awards for **best paper**, **2x best paper candidate** and **2x best artifact**
  ‡ includes four **single author** publications (three full length papers and one short paper)
  † was also invited to give three additional talks, but had to decline

List of Publications by Category
(* = grad student, **bold** = my student)

Under Submission or Pending Submission

1. **Privacy Preserving Vertical K-means Clustering.**
   Federico Mazzone*, Florian Kerschbaum and Trevor Brown. 14 pages.
   (I do not yet know what the author order will be.)

2. **Persistent HyTM via Fast Path Fine-Grained Locking.**
   Guy Coccimiglio*, Trevor Brown and Srivatsan Ravi. 14 pages.

3. **Cost of concurrency in hybrid transactional memory.**
   Trevor Brown and Srivatsan Ravi. 39 pages.

4. **CLpush: Fine-Grained Cache Control for Concurrent Programs.**
   Gautam Pathak*, **Ajay Singh***, Trevor Brown, Ali Mashtizadeh. 13 pages.

Books

5. **Shared Memory Synchronization (2e).**
   Springer Synthesis Lectures on Computer Arch. 253 pages. Published 2024.
   ISBNs: 978-3-031-38683-1, 978-3-031-38686-2 and 978-3-031-38684-8.
   Michael Scott and Trevor Brown. [Link to Springer page]

Journal Publications

6. **Simple, Fast and Widely Applicable Concurrent Memory Reclamation via Neutralization.** [Paper]
   Ajay Singh*, Trevor Brown and Ali Mashtizadeh. 18 pages.
   IEEE Transactions on Parallel and Distributed Computing (**TPDS**) Feb.'24, vol.35, pp 203-220. (3.757 impact factor 2023-2024)

7. **PHyTM: persistent hybrid transactional memory.**
   Hillel Avni and Trevor Brown. [Paper] [Slides]
   Very Large Data Bases Journal (**VLDBJ'16 / VLDB'16**) Volume 10(4), pp 409-420.
   (17% acceptance; CORE A*-rank, ERA A-rank.)
   Also appeared at the 43rd Int. Conference VLDB'17.

Conference Publications

8. **More Bang for Your Buck(et): Fast and Space-efficient Hardware-accelerated Coarse-granular Indexing on GPUs.** [Preprint]
   Justus Henneberg*, Felix Schuhknecht, **Rosina Kharal*** and Trevor Brown.
   41st IEEE Intl. Conf. on Data Engineering (**ICDE'25**), 14 pages.
   (CORE A*-rank; Qualis A1-rank, ERA A-rank)

9. **Publish on Ping: A Better Way to Publish Reservations in Memory Reclamation for Concurrent Data Structures.** [Code]

Ajay Singh* and Trevor Brown.
30th ACM Symp. on Princ. And Pract. of Parallel Programming (**PPoPP'25**), 12 pages.
(20.1% acceptance; Qualis A2-rank, ERA A-rank)

10. **Are Your Epochs Too Epic? Batch Free Can Be Harmful.**
Daewoo Kim*, Trevor Brown, **Ajay Singh***. [Paper] [Slides] [Code]
29th ACM Symp. on Princ. And Pract. of Parallel Programming (**PPoPP'24**), 12 pages.
(20.9% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)

11. **Practical Hardware Transactional vEB Trees.** [Paper] [Slides] [Code]
Mohammad Khalaji*, Trevor Brown, Khuzaima Daudjee, Vitaly Aksenov.
29th ACM Symp. on Princ. And Pract. of Parallel Programming (**PPoPP'24**), 12 pages.
(20.9% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)

12. **The fence complexity of persistent sets.**
Gaetano Coccimiglio*, Trevor Brown and Srivatsan Ravi. [Paper] [Slides] [Code]
25th Intl. Symp. on Stabilization, Safety, and Security of Distr. Sys. (**SSS'23**), pp 36-51.
(40% acceptance; CORE/ERA C-rank)

13. **Efficient Hardware Primitives for Immediate Memory Reclamation in Optimistic Data Structures.**
Ajay Singh*, Trevor Brown and Michael Spear. [Paper] [Slides] [Code]
37th IEEE International Parallel and Distributed Processing Symposium (**IPDPS'23**).
(stats not yet released---26% accept in 2022; CORE A-rank, Qualis A1-rank, ERA A-rank)

14. **Performance Anomalies in Concurrent Data Structure Microbenchmarks.**
Rosina Kharal* and Trevor Brown. [Paper] [Slides]
26th International Conf. on Principles of Distributed Systems (**OPODIS'22**), 23 pages.
(32% acceptance; CORE B-rank, Qualis B2-rank, ERA B-rank)

15. **PREP-UC: A practical replicated persistent universal construction.**
Gaetano Coccimiglio*, Trevor Brown and Srivatsan Ravi. [Paper] [Slides] [Code]
34th ACM Symposium on Parallelism in Algorithms and Architectures (**SPAA'22**), 13 pages.
(30% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)
Best paper finalist (one of three).

16. **Elimination (a,b)-trees with fast, durable updates.**
Anubhav Srivastava* and Trevor Brown. [Paper] [Video] [Artifact] [Code]
27th ACM Symp. on Princ. And Practice of Parallel Programming (**PPoPP'22**), 12 pages.
(accepted w/o conditions - 18.6% of papers; CORE A-rank, Qualis A2-rank, ERA A-rank)

17. **PathCAS: an efficient middle ground for concurrent search data structures.**
Trevor Brown, **William Sigouin*** and Dan Alistarh. [Paper] [Slides] [Video] [Code]
27th ACM Symp. on Princ. And Practice of Parallel Programming (**PPoPP'22**), 12 pages.
(24.6% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)
Received the best artifact award.

18. **NBR: Neutralization based reclamation.**
Ajay Singh*, Trevor Brown and Ali Mashtizadeh. [Paper] [Slides] [Video] [Artifact] [Repo]
26th ACM Symp. on Princ. and Practice of Parallel Programming (**PPoPP'21**), 16 pages.

(17% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank.)
Best paper candidate (top 4 papers).
Received the best artifact award [ACM announcement tweet] [ACM DL page].

19. **Memory tagging: minimalist synchronization for scalable concurrent data structures.**
Dan Alistarh, Trevor Brown and Nandini Singhal. [Paper] [Slides] [Video]
32nd ACM Symp. on Parallelism in Algorithms and Architectures (**SPAA'20**), pp 37-49.
(32% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)

20. **Non-blocking interpolation search trees with doubly-logarithmic running time.**
Trevor Brown, Aleksandar Prokopec and Dan Alistarh. [Paper] [Slides] [Audio] [Code]
25th ACM Symp. On Princ. and Pract. of Parallel Programming (**PPoPP'20**), pp 4:(1-16)
(23% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)
Received the best paper award.

21. **Snapshot-based synchronization: a fast replacement for hand-over-hand locking.**
Eran Gilad*, Trevor Brown, Mark Oskin and Yoav Etsion. [Paper]
24th Intl. European Conf. on Parallel and Distr. Computing (**EUROPAR'18**), pp 465-479.
(29% acceptance; Qualis A2-rank, ERA A-rank)

22. **Relaxed schedulers efficiently parallelize sequential algorithms.**
Dan Alistarh, Trevor Brown, Justin Kopinsky and Giorgi Nadiradze. [Paper]
37th ACM Symp. on the Principles of Distributed Computing (**PODC'18**), pp 377-386.
(25% acceptance; CORE A*-rank, Qualis A1-rank, ERA A-rank)

23. **Distributionally linearizable data structures.**
Dan Alistarh, Trevor Brown, Justin Kopinsky*, Giorgi Nadiradze* and Jerry Li*. [Paper]
30th ACM Symp. on Parallelism in Algorithms and Architectures (**SPAA'18**), pp 133-142.
(30% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)

24. **Getting to the root of concurrent search tree performance.**
Maya Arbel-Raviv*, Trevor Brown and Adam Morrison. [Paper] [Slides] [Audio]
2018 USENIX Annual Technical Conference (**USENIX ATC'18**), pp 295-306.
(20% acceptance; CORE A-rank, Qualis A1-rank, ERA A-rank)

25. **Harnessing epoch-based reclamation for efficient range queries.**
Maya Arbel-Raviv* and Trevor Brown. [Paper] [Slides] [Code]
23rd ACM Symp. on Princ. and Practice of Parallel Programming (**PPoPP'18**), pp 14-27.
(20% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank.)

26. **Reuse, don't recycle: transforming lock-free algorithms that throw away descriptors.**
Maya Arbel-Raviv* and Trevor Brown*. [Paper] [Slides] [Video] [Code]
31st EATCS Symposium on Distributed Computing (**DISC'17**), pp 4:(1-16).
(24% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)

27. **Cost of concurrency in hybrid transactional memory.**
Trevor Brown* and Srivatsan Ravi. [Paper] [Slides] [Code]
31st EATCS Symposium on Distributed Computing (**DISC'17**), pp 9:(1-16).
(24% acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)

28. **A template for implementing fast lock-free trees using HTM.**
    Trevor Brown*. [Paper] [Slides] [Code]
    36th ACM Symp. on the Principles of Distributed Computing (**PODC'17**), pp 293-302.
    (25% acceptance; CORE A*-rank, Qualis A1-rank, ERA A-rank)

29. **Investigating the performance of hardware transactions on a multi-socket machine.**
    Trevor Brown*, Alex Kogan, Yossi Lev and Victor Luchangco. [Paper] [Slides]
    28th ACM Symp. on Parallelism in Algorithms and Architectures (**SPAA'16**), pp 121-132.
    (24% acceptance in 2017. 2016 unknown. CORE A-rank, Qualis A2-rank, ERA A-rank.)

30. **Reclaiming memory for lock-free data structures: there has to be a better way.**
    Trevor Brown*. [Paper] [Slides] [Code]
    34th ACM Symp. on the Principles of Distributed Computing (**PODC'15**), pp 261-270.
    (24% acceptance; CORE A*-rank, Qualis A1-rank, ERA A-rank)

31. **B-slack trees: space efficient B-trees.**
    Trevor Brown*. [Paper] [Slides] [Code]
    14th Scandinavian Symp. and Workshops on Algorithm Theory (**SWAT'14**), pp 122-133.
    (24% acceptance; CORE B-rank, Qualis B1-rank, ERA B-rank)

32. **A general technique for non-blocking trees.**
    Trevor Brown*, Faith Ellen and Eric Ruppert. [Paper] [Slides] [Code]
    19th ACM Symp. on Princ. and Practice of Parallel Programming (**PPoPP'14**), pp 329-342.
    (15% paper acceptance; CORE A-rank, Qualis A2-rank, ERA A-rank)

33. **Pragmatic primitives for non-blocking data structures.**
    Trevor Brown*, Faith Ellen and Eric Ruppert. [Paper] [Slides] [Code]
    31st ACM Symposium on the Principles of Distributed Computing (**PODC'13**), pp 13-22.
    (23% paper acceptance; CORE A*-rank, Qualis A1-rank, ERA A-rank)

34. **Range queries in non-blocking k-ary search trees.**
    Trevor Brown* and Hillel Avni*. [Paper] [Code]
    16th International Conf. on Principles of Distributed Systems (**OPODIS'12**), pp 31-45.
    (27% acceptance; CORE B-rank, Qualis B2-rank, ERA B-rank)

35. **Non-blocking k-ary search trees.**
    Trevor Brown* and Joanna Helga*. [Paper] [Slides] [Video] [Code]
    15th International Conf. on Principles of Distributed Systems (**OPODIS'11**), pp 207-221.
    (CORE B-rank, Qualis B2-rank, ERA B-rank) [During undergrad]

Workshop Papers

36. **Cost of concurrency in hybrid transactional memory.**
    Trevor Brown* and Srivatsan Ravi. [Paper] [Slides]
    12th ACM SIGPLAN Workshop on Transactional Computing (**TRANSACT'17**), 8 pages.
    (Competitive peer-review, but no statistics.)
    Preliminary version of the conference paper at DISC'17.

37. **Persistent hybrid transactional memory.**
    Hillel Avni* and Trevor Brown*. [Paper] [Slides]

11th ACM SIGPLAN Workshop on Transactional Computing (**TRANSACT'16**), 8 pages.
(Competitive peer-review, but no statistics.)
Preliminary version of the conference paper at VLDB'17.

38. **Investigating the performance of hardware transactions on a multi-socket machine.**
Trevor Brown*, Alex Kogan, Yossi Lev and Victor Luchangco. [Paper] [Slides]
11th ACM SIGPLAN Workshop on Transactional Computing (**TRANSACT'16**), 8 pages.
(Competitive peer-review, but no statistics.)
Preliminary version of the conference paper at SPAA'16.

Short Papers and Posters

39. **Unexpected scaling in path copying trees.**
Vitaly Aksenov, Trevor Brown, Alexander Fedorov* and Ilya Kokorin*. [Paper] [Slides]
28th ACM Symp. on Princ. And Practice of Parallel Programming (**PPoPP'23**).
(23.6% acceptance; Qualis A2-rank, ERA A-rank)
Poster and short paper (3 pages).

40. **Brief Announcement: Performance Anomalies in Concurrent Data Structure Microbenchmarks.**
Rosina Kharal* and Trevor Brown. [Paper] [Slides]
36th EATCS Symposium on Distributed Computing (**DISC'22**).
(Qualis A2-rank, ERA A-rank)
Short version (3 pages) of conference paper at OPODIS'22.

41. **Reuse, don't recycle: transforming lock-free algorithms that throw away descriptors.**
Maya Arbel-Raviv* and Trevor Brown*. [Paper] [Poster] [Code]
22nd ACM Symp. on Princ. and Practice of Parallel Programming (**PPoPP'17**), pp 429-430.
(22% acceptance; Qualis A2-rank, ERA A-rank)
Poster and short version of conference paper at DISC'17.

42. **Concurrent data structures.**
Faith Ellen and Trevor Brown*. [Paper] [Slides]
35th ACM Symp. on the Principles of Distributed Computing (**PODC'16**), pp 151-153.
Short paper to accompany an invited talk by Faith Ellen at PODC'16.

43. **Faster data structures in transactional memory using three paths.**
Trevor Brown*. [Paper] [Slides] [Code]
29th EATCS Symposium on Distributed Computing (**DISC'15**), pp 671-672.
(31% acceptance; Qualis A2-rank, ERA A-rank)
Short version of conference paper at PODC'17.

44. **A general technique for non-blocking trees.**
Trevor Brown*, Faith Ellen and Eric Ruppert. [Paper] [Slides] [Code]
27th EATCS Symposium on Distributed Computing (**DISC'13**), pp 567-568.
(26% acceptance; Qualis A2-rank, ERA A-rank)
Short version of conference paper at PPoPP'14.

Articles and Technical Reports

45. **Are Your Epochs Too Epic? Batch Free Can Be Harmful.**
Daewoo Kim*, Trevor Brown, **Ajay Singh***. [Paper]
ArXiv Computing Research Repository (CoRR), abs/2401.11347, 33 pages.
Full version of (12 page) conference paper at PPoPP'24. Adds additional background (+1 page), additional data structures to experiments (+3 pages), additional machines to experiments (+2 pages), and additional allocators to experiments (+14 pages).

46. **Efficient Hardware Primitives for Immediate Memory Reclamation in Optimistic Data Structures (2023). Ajay Singh***, Trevor Brown and Michael Spear.
ArXiv Computing Research Repository (CoRR), abs/2302.12958. 13 pages.
Full version of (11 page) conf. paper at IPDPS'23. Adds proof of correctness (+1.5 pages).

47. **PathCAS: An efficient middle ground for concurrent search data structures (2022).**
Trevor Brown, **William Sigouin*** and Dan Alistarh. [Paper]
ArXiv Computing Research Repository (CoRR), abs/2212.09851, 34 pages.
*Full version of (12 page) conference paper at PPoPP'22. **Adds** algorithm implementation details (+1 page), a new AVL tree algorithm (+4 pages), a new dynamic connectivity algorithm (+4 pages), a detailed description of a bug in a published algorithm (+2.5 pages), and additional experiments (+6.5 pages).*

48. **Performance anomalies in concurrent data structure microbenchmarks (2022).**
Rosina Kharal* and Trevor Brown. [Paper]
ArXiv Computing Research Repository (CoRR), abs/2208.08469, 27 pages.
*Full version of (16 page) conference paper at OPODIS'22. **Adds** experiments (+3 pages).*

49. **Elimination (a,b)-trees with fast, durable updates (2022).**
Anubhav Srivastava* and Trevor Brown. [Paper]
ArXiv Computing Research Repository (CoRR), abs/2112.15259, 22 pages.
*Full version of conference paper at PPoPP'22. **Expanded** proofs (+5 pages) and experiments (+3 pages).*

50. **NBR: neutralization based reclamation (2021).**
Ajay Singh*, Trevor Brown and Ali Mashtizadeh. [Paper]
ArXiv Computing Research Repository (CoRR), abs/2012.14542, 22 pages.
*Full version of (16 page) conference paper at PPoPP'21. **Expanded** proofs (+1 page), discussion of applicability of our techniques (+3 pages), experiments (+3 pages).*

51. **Analysis and evaluation of non-blocking interpolation search trees (2020).**
Aleksandar Prokopec, Trevor Brown and Dan Alistarh. [Paper]
ArXiv Computing Research Repository (CoRR), abs/2001.00413, 17 pages.
Correctness and complexity proofs and different experiments for PPoPP'20 paper.

52. **On the cost of concurrency in hybrid transactional memory (2019).**
Trevor Brown and Srivatsan Ravi. [Paper]
ArXiv Computing Research Repository (CoRR), abs/1907.02669, 17 pages.
*Full version of (10 page) conference paper at DISC'17. **Adds 4 pages of proofs.***

53. **Reuse, don't recycle: transforming lock-free algo. that throw away descriptors (2017).**
Maya Arbel-Raviv* and Trevor Brown*. [Paper]
ArXiv Computing Research Repository (CoRR), abs/1708.01797, 32 pages.
Full version of (16 page) conference paper at DISC'17. ***Expanded*** *description of algorithms (+8 pages), added proofs (+4 pages), added experiments (+4 pages).*

54. **A template for implementing fast lock-free trees using HTM (2017).**
Trevor Brown*. [Paper]
ArXiv Computing Research Repository (CoRR), abs/1708.04838, 20 pages.
Full version of (10 page) conference paper at PODC'17. ***Expanded*** *intro (+1 page), background (+1 page), algorithms (+6 pages), additional applications (+2 pages).*

55. **Techniques for constructing efficient lock-free data structures (2017).**
PhD thesis, University of Toronto. [Paper]
Committee: Faith Ellen (University of Toronto), Azadeh Farzan (University of Toronto), Vassos Hadzilacos (University of Toronto), Maurice Herlihy (Brown University), Ryan Johnson (University of Toronto), Sam Toueg (University of Toronto).
ArXiv Computing Research Repository (CoRR), abs/1712.05406, 312 pages.

56. **Reclaiming memory for lock-free data structures: there has to be a better way (2017).**
Trevor Brown*. [Paper]
ArXiv Computing Research Repository (CoRR), abs/1712.01044, 27 pages.
Full version of (10 page) conference paper at PODC'15. ***Expanded*** *related work (+4 pages), additional details in algorithms, implementation and record manager.*

57. **B-slack trees: space efficient B-trees (2017).**
Trevor Brown*. [Paper]
ArXiv Computing Research Repository (CoRR), abs/1712.05020, 19 pages.
Full version of (12 page) conference paper at SWAT'14. ***Adds 7 pages of proofs.***

58. **A general technique for non-blocking trees (2017).**
Trevor Brown*, Faith Ellen and Eric Ruppert. [Paper]
ArXiv Computing Research Repository (CoRR), abs/1712.06687, 41 pages.
Full version of (10 page) conference paper at PPoPP'14. ***Adds 23 pages of proofs.***

59. **Pragmatic primitives for non-blocking data structures (2017).**
Trevor Brown*, Faith Ellen and Eric Ruppert. [Paper]
ArXiv Computing Research Repository (CoRR), abs/1712.06688, 47 pages.
Full version of (10 page) conference paper at PODC'13. ***Adds 34 pages of proofs.***

60. **Range queries in non-blocking k-ary search trees (2017).**
Trevor Brown* and Hillel Avni*. [Paper]
ArXiv Computing Research Repository (CoRR), abs/1712.05101, 18 pages.
Full version of (15 page) conference paper at OPODIS'12. ***Adds 3 pages of proofs.***

61. **Non-blocking k-ary search trees (2017).**
Trevor Brown* and Joanna Helga*. [Paper]
York University Technical Report CSE-2011-04, 52 pages.
Full version of (16 page) conference paper at OPODIS'11. ***Adds 36 pages of proofs.***

Patents

62. **Adaptive techniques for improving performance of hardware transactions on multi-socket machines.** Oracle Labs, US patent 10,127,088. 34 pages. Granted Nov 2018. Alex Kogan, Yossi Lev, Victor Luchangco and Trevor Brown*. [Link]

Talks (C = conference; S = seminar; I = invited; L = guest lecture)

I was invited for a 1h talk at the 1st Workshop on Highlights of Asynchronous Concurrent and Distributed Algorithms (HACDA'23). Unfortunately, I had to **decline** due to timing. (I was also **invited** for HACDA'24, EMERALD'24, and Huawei's 2023 Software Summit, but had to **decline**.)

Unexpected scaling in path copying trees.
- [C] 28th ACM Symp. on Princ. and Practice of Parallel Programming, Montreal, Canada (Feb 2023). [Slides]

Fast and simple concurrent data structures using PathCAS.
- [I] Invited talk (40 minutes) for the University of Windsor, Computer Science Colloquium Series. University of Windsor, Ontario, Canada (Dec 2022). [Slides]

PathCAS: An efficient middle ground for concurrent search data structures.
- [C] 27th ACM Symposium on Principles and Practice of Parallel Programming, Seoul, South Korea (virtual, Apr 2022). [Video]

The fence complexity of persistent sets.
- [S] 45 minute presentation at the 3rd Waterloo-Huawei Joint Innovation Workshop (311 registrants), Waterloo, Canada (Jun 2021).

Memory tagging: minimalist synch. for scalable concurrent data structures.
- [C] 32nd ACM Symposium on Parallel Algorithms and Architectures, *Virtual conference* (July 2020). [Slides] [Video]

Non-blocking interpolation search trees with doubly-logarithmic running time.
- [C] 25th ACM Symposium on Principles and Practice of Parallel Programming, San Diego, United States (Feb 2020). ***Best paper award.*** [Slides] [Audio]

I was invited for a 1h talk at the Hydra'20 and Hydra'22 conferences on concurrent and distributed computing**.** However, I had to decline due to timing, and war, resp.

Scalable infrastructure for next-generation data management systems.
- [S] 45 minute presentation at the 2nd Waterloo-Huawei Joint Innovation Workshop (164 registrants), Waterloo, Canada (Jun 2020).

Practical aspects of multicore programming.
- [I] Invited talk at the 2nd Summer School on the Practice and Theory of Distributed Computing (200+ attendees), St. Petersburg, Russia (Jul 2019). ***3 hours*.**
Rated 2nd best out of 9 invited talks by highly distinguished speakers, including a Turing award winner. [Talk] [Speakers list]

Getting to the root of concurrent search tree performance.
- **[C]** 2018 USENIX Annual Technical Conference, Boston, United States (Jul 2018). [Slides] [Audio]

Getting to the root of concurrent BSTs.
- **[L]** TU Wien, Vienna, Austria (June 2018). 90 minute guest lecture. [Slides]

Towards correct and efficient multicore programming.
- **[S]** University of Waterloo, Waterloo, Canada (Mar 2018). [Slides]
- **[S]** University of California Santa Cruz, California, United States (Feb 2018).
- **[S]** Simon Fraser University, Burnaby, Canada (Feb 2018).
- **[S]** University of Waterloo, Waterloo, Canada (Jan 2018).

Good data structure experiments are R.A.R.E.
- **[I]** Invited talk at the 1st Workshop on the Theory and Practice of Concurrency, Vienna, Austria (Oct 2017). *Held in conjunction with DISC'17.* 60min. [Slides] [Video]
- **[S]** Oath/Yahoo! Labs, Haifa, Israel (Sep 2017). 60min. [Slides]

Reuse, don't recycle: transforming lock-free algorithms that throw away descriptors.
- **[C]** 31st ACM Symposium on Distributed Computing, Vienna, Austria (Oct 2017). [Slides] [Video]
- **[S]** Riot Games, Los Angeles, California (Nov 2019) – video talk

Cost of concurrency in hybrid transactional memory.
- **[C]** 31st ACM Symposium on Distributed Computing, Vienna, Austria (Oct 2017). [Slides]

A template for implementing fast lock-free trees using HTM.
- **[C]** 36th ACM Symposium on the Principles of Distributed Computing, Washington, United States (Jul 2017). [Slides]

Techniques for Constructing Efficient Lock-free Data Structures.
- **[S]** Technion - Israel Institute of Technology, Haifa, Israel (May 2017). 60min. [Slides]
- **[S]** University of Toronto, Toronto, Canada (Mar 2017). 60min. [Slides]

Investigating the perf. of hardware transactions on a multi-socket machine.
- **[C]** 28th ACM Symposium on Parallelism in Algorithms and Architectures, Monterey, United States (Jul 2016). [Slides]
- **[C]** 11th ACM SIGPLAN Workshop on Transactional Computing, Barcelona, Spain (Mar 2016). [Slides]

Persistent hybrid transactional memory.
- **[C]** 11th ACM SIGPLAN Workshop on Transactional Computing, Barcelona, Spain (Mar 2016). [Slides]

Faster data structures in transactional memory using three paths.
- **[C]** 29th ACM Symposium on Distributed Computing, Tokyo, Japan (Oct 2015). [Slides]

Scalable transactions on NUMA systems.
- **[S]** Oracle Labs East, Burlington, United States (Aug 2015). [Slides]

Reclaiming memory for lock-free data structures: there has to be a better way.
- **[C]** 34th ACM Symposium on the Principles of Distributed Computing, San Sebastien, Spain (Jul 2015). [Slides]

Hardware transactional memory and the lemming effect.
- **[S]** University of Toronto, Toronto, Canada (Apr 2015). [Slides]

Java Experiments on MTL: From past mistakes to best practices.
- **[L]** York University, Toronto, Canada (Mar 2015). 90 minute guest lecture. [Slides]

Memory reclamation for lock-free data structures.
- **[S]** University of Toronto, Toronto, Canada (Aug 2014).

B-slack trees: space efficient B-trees.
- **[C]** 14th Scandinavian Symposium and Workshops on Algorithm Theory, Copenhagen, Denmark (Jun 2014). [Slides]

A general technique for non-blocking trees.
- **[C]** 19th ACM Symposium on Principles and Practice of Parallel Programming, Orlando, United States (Feb 2014). [Slides]
- **[C]** 27th ACM Symposium on Distributed Computing, Jerusalem, Israel (Aug 2013). [Slides]

Pragmatic primitives for non-blocking data structures.
- **[C]** 31st ACM Symposium on the Principles of Distributed Computing, Montreal, Canada (Jul 2013). [Slides]
- **[S]** University of Toronto, Toronto, Canada (Jul 2013).

Building a non-blocking chromatic tree.
- **[S]** TransForm School on Research Directions in Distributed Computing, Heraklion, Crete (Jun 2013). [Slides]

Range queries in non-blocking k-ary search trees.
- **[C]** 16th International Conference on Principles of Distributed Systems, Rome, Italy (Nov 2012).

Non-blocking k-ary search trees.
- **[C]** 15th International Conference on Principles of Distributed Systems, Toulouse, France (Oct 2011). [Slides] [Video]

Experiences with Intel's Multicore Testing Lab.
- **[L]** York University, Toronto, Canada (2011). 60 minute guest lecture.

Software Distributions

*In the following, "LOC" refers to non-blank, non-whitespace, non-comment lines of code. When one software artifact includes (parts of) another, I give my best estimate of the **new** code written for the project.*

- **C++ publish-on-ping (POP) memory reclamation library and benchmark (2024). [2,200 new LOC].** Uses POSIX signals to accelerate pointer-reservation based memory reclamation techniques like hazard pointers and hazard eras, improving the state-of-the-art in concurrent data structures. The artifact is for a corresponding PPoPP'25 paper.
  Artifact: http://popartifact.tbrown.pro
  Library: http://pop.tbrown.pro
- **C++ Persistent Hybrid Transactional Memory via Fast Path Fine-Grained Locking (2024). [3,200 new LOC].** A new transactional memory library that allows users to wrap single threaded volatile code to be automatically executed by multiple software and hardware threads (using Intel RTM) and made persistent (using Intel DCPMM non-volatile RAM). Sample data structures included.
  http://lockphytm.tbrown.pro
- **C++ hardware transactional vEB trees (2024). [4,500 new LOC].** A new concurrent ordered-set/map supporting fast, doubly-logarithmic successor/predecessor queries. Outperforms the state of the art in uniform and moderately skewed workloads. Software artifact for corresponding PPoPP'24 paper.
  http://vebartifact.tbrown.pro
- **C++ neutralization based memory reclamation algorithms and benchmark (2023). [2,500 new LOC].** To our knowledge, this is the most comprehensive benchmarking of safe memory reclamation algorithms conducted to date. Implementations of novel neutralization based algorithms are included. Benchmarks include concurrent lists, hash tables and search trees. Artifact corresponds to TPDS'23 paper above.
  Artifact: http://nbrartifact.tbrown.pro
  Library: http://nbrrepo.tbrown.pro
- **Simulation of novel Conditional Access instructions for MESI & MOESI coherence protocols (2023). [28,000 LOC; 10,000 new (4,900 new in the simulator)].** Our Conditional Access work at IPDPS'23 introduced novel conditional load/store instructions for hardware-software codesign of concurrent data structures algorithms. This is a simulation of those instructions, with accompanying benchmarks, built on the MIT Graphite tiled-multicore simulator.
  http://ca.tbrown.pro
- **C++ persistent concurrent linked-list based sets and benchmarks (2023). [2,500 new LOC].** A library of several different persistent concurrent sets which each satisfy different correctness criteria. Artifact for the corresponding SSS'23 conference paper.
  http://fcomplexity.tbrown.pro
- **PREP-UC: universal construction library in C++ (2022). [3,000 new LOC].** Automatically transforms single threaded code into multithreaded, NUMA friendly, durable linearizable code (for Intel DCPMM non-volatile RAM). Software artifact for a corresponding SPAA'22 conference paper.
  http://prepuc.tbrown.pro

- **PathCAS based data structures in C++ (2022). [8,000 new LOC].**
  A library for simple and fast lockless synchronization, along with many example data structures that use it, and benchmarks to compare them against leading competitors. Software artifact for a corresponding PPoPP'22 paper. This won **best artifact**
  Artifact: http://pathcasartifact.tbrown.pro
  Library: http://pathcas.tbrown.pro
- **Elimination (a,b)-trees in C++ (2022). [5,000 new LOC].**
  A new concurrent ordered-set/ordered-map data structure with extremely high performance under high update workloads (substantially outperforming the state of the art). Software artifact for a corresponding PPoPP'22 conference paper.
  Artifact: http://pubelimartifact.tbrown.pro
  Library: http://pubelim.tbrown.pro
- **TMLib C++ transactional library (2021 – present) [12,700 LOC; 10,000 new].**
  A library of ten transactional memory algorithms which allow a programmer to specify that certain blocks of code should be run atomically as transactions. I implemented nine of the algorithms myself. To my knowledge, this is the most complete repository of transactional memory algorithms available.
  http://tmlib.tbrown.pro
- **TMBench: C++ transactional data structure benchmark (2021 – present) [88,000 LOC; 30,000 new].** An extension of SetBench for comparing the performance of different implementations of transactional memory. It uses each of the transactional memory algorithms in TMLib to implement a variety of set/map data structures, and evaluates these resulting data structures in microbenchmarks and database workloads.
  http://tmbench.tbrown.pro
- **Setbench: C++ data structure test harness & benchmark (2018 - present) [57,000 LOC; 10,000 new].** A rigorous performance benchmark for concurrent set/map data structures. Provides tools for avoiding common experimental mistakes. Fixes errors in publicly available concurrent sets. Includes extensive tooling for running, validating and explaining experimental results.
  http://setbench.tbrown.pro
- **Support for range query operations in C++ (2018) [55,000 LOC; 25,000 new].**
  Three novel algorithms for adding range query operations to existing data structures. Seven concurrent data structures augmented with range query support (producing up to five variants of each data structure). Includes in-memory DB benchmark (DBx1000).
  http://range-queries.tbrown.pro
- **Reusable descriptors for lock-free data structures in C++ (2018) [29,000 LOC]**
  A library for accelerating advanced lock-free data structures. Demonstrated with four increasingly complex data structures.
  http://weak-descriptors.tbrown.pro
- **LLX/SCX primitives for C++ (2018).**
  This is the state-of-the-art implementation of the LLX and SCX synchronization primitives, with many improvements over the original.
  http://scx.tbrown.pro
- **C++ hardware transactional lock-free data structure library (2017) [23,000 LOC].**
  The first C/C++ implementation of the LLX and SCX synchronization primitives.

Provides unbalanced binary search trees and relaxed (a,b)-trees using these primitives, as well as eight different transactional memory based data structures.
http://3path-htm.tbrown.pro

- **C++ lock-free memory reclamation library (2015) [9,000 LOC].**
  Provides a record manager library with allocation, reclamation and object pooling plugins (including five allocators and four memory reclamation algorithms).
  Implemented lock-free BSTs and Chromatic trees in C++ using this library.
  http://debra.tbrown.pro
- **Java lock-free data structure library (2014) [14,600 LOC; 6,600 new].**
  Introduced the first lock-free unbalanced binary search tree implementation, the first implementation of LLX and SCX synchronization primitives, and the first lock-free implementations of k-ary search trees, relaxed AVL trees, Chromatic trees, b-slack trees.
  Experimental test harness carefully integrates *thirteen* competing data structures.
  http://java.tbrown.pro

- Estimated LOC *personally* contributed: 156,000
  (estimate includes *some* software distributions, and **excludes** many [Code] links above)

Teaching

- *Asst. Professor – University of Waterloo* (2018 – present)

F24	CS341	algorithms (two sections, ~160 students – **4.3**/5 avg. evaluation)
W24	CS798	multicore programming, ~30 students, **4.9**/5 evals.)
F23	CS341	algorithms (two sections, ~160 students – **4.2**/5 avg. evaluation)
W23	CS798	multicore programming, 30 students, **4.8**/5 evals.
S22	CS341	algorithms (two sections, ~110 students – **4.8**/5, **4.3**/5 evals.)
F21	CS341	algorithms (two sections, ~200 students – **4.5**/5 avg. evaluation)
W21	CS798-043	multicore programming, 26 students

  - course generated high interest: 32 initially enrolled, plus 36 waitlisted
  - piazza participation: 1533 student posts (59 per student!), 255 by instructor
  - avg teaching evaluation of **4.7**/5

W20	CS341	algorithms (two sections, 177 students, **4.6**/5 eval. in section 1)
F19	CS798-003	multicore programming (18 students, **4.9**/5 average evaluation)
W19	CS341	algorithms (87 students, **4.3**/5 average evaluation)
F18	CS798	multicore programming (20 students, **4.8**/5 average evaluation)

  Reading Group attended by up to 15 students from multiple research groups

- *Teaching Assistant – University of Toronto* (2011 – 2014)

CSC2221	(graduate) theory of distributed computing (2013)
CSC263	data structures and analysis (2011, 2012, 2013, 2014) - Gave approximately **60 lectures**, each an hour long - 2014 teaching evaluation average was 4.9+ out of 5
CSC369	principles of operating systems (2012)
CSC265	enriched data structures and analysis (2011)

- *Programming contest coach – York University* (2009)

Curriculum Development

- **CS798 – Multicore Programming**
  New lecture based graduate course. Extremely highly regarded by students.
  Could be adapted for undergrad, to follow CS343 Concurrent & Parallel Programming.
  Designed 18 x 90 min brand new lectures, with detailed slides and animations.
  27h of video lectures recorded with a live audience (before COVID!) - Example video
  Designed seven lab-style programming assignments and supporting Marmoset scripts.
  Wrote 3,000+ lines of starter code.
- **CS341 – Introduction to Algorithms**
  Designed new slides packed with animations
  Recorded 23 hours of high-quality video lectures (~300 hours of prep.)
  - Example lecture video - bit.ly/3MXMoTC

Research Funding (My share**: $965,755.00**)

NSERC Collaborative Research and Development ($209,700) 2020 – 2023
Trevor Brown (75%) and Ali Mashtizadeh (25%)
I **declined** additional NSERC CRD COVID relief funds.

Huawei Research Gift ($120,000) 2021 – 2022

Huawei Waterloo Joint Innovation Grant ($190,000) 2019 – 2021
Trevor Brown (75%) and Ali Mashtizadeh (25%)

Ontario Research Fund: Research Infrastructure ($101,000) 2020 – 2021
Trevor Brown (25%) and Ali Mashtizadeh (75%)

CFI John Evans Leaders Fund Grant ($101,000) 2020 – 2021
Trevor Brown (25%) and Ali Mashtizadeh (75%)

NSERC Discovery Grant ($234,000) [39k/y] 2019 – 2025

NSERC Discovery Launch Supplement ($12,500) 2019 – 2023

Waterloo Startup Grant ($120,000) 2018 – 2022

NSERC Postdoctoral fellowship ($90,000) 2017 – 2018
Second highest ranked applicant in Canada

$200,000+ of scholarships & awards prior to Waterloo omitted.

Summary of Supervision

	Sole Supervisor	**Co-Supervised**
Current Masters		
Graduated Masters	5	
Current PhD		3 (2 effectively sole supervised)
Graduated PhD	1	
Current PDF	1	
Completed PDF		
Lifetime RAs	6	1

Thesis Supervision

- Mohammad Khalaji: PhD** (UW, W2022 – present)
- Gaetano Coccimiglio: PhD* (UW, F2021 – present)
- Rosina Kharal: PhD* (UW, F2019 – present)
- Ajay Singh: PhD* (UW, F2019 – F24). Joined ICS Forth for Postdoc.
- Gautam Pathak: MMath (UW, F2021 – S2023). Joined ARM.
  Thesis: CLPush: Proactive Cache Placement in NUMA Applications
- Daewoo Kim: MMath (UW, F2020 – W2023). Continued to PhD.
  Thesis: Understanding NUMA Effects on Memory Allocation and Reclamation
- Anubhav Srivastava: MMath (UW, S2020 – S2021). Joined Yugabyte.
  Thesis: Extremely Fast (a,b)-trees at All Contention Levels
- Gaetano Coccimiglio: MMath (UW, F2019 – S2021). Continued to PhD.
  Thesis: The Fence Complexity of Persistent Sets
- William Sigouin: MMath (UW, W2018 – W2020). Joined Workiva.
  Thesis: Concurrent Data Structures Using Multiword Compare and Swap
- Informal supervision prior to Waterloo omitted

*Co-supervised (as required by UW) with Peter Buhr, **with Khuzaima Daudjee

Other Student Supervision (W = Jan-Apr, S=May-Aug, F=Sep-Dec)

- Tom Iagovet intern (UW, W24 – present).
- Sean Ovens PDF (UW, F23 – present).
- Alexander Fedorov: intern (UW & IST Austria, W23 – W24). PhD @ IST.
- Gaurav Gupta: intern (USC, S21 – S21).
- Abhirup Das: intern (UW, F20 – W21). Joined Cisco Systems.
- Gautam Pathak: intern (UW, W20 – S20). Joined for an MMath.
- Daewoo Kim: intern (UW, W19 – S19). Joined for an MMath.

Thesis Examination

- **Thesis committee member / reader** for
  - Jingbang Chen (PhD, University of Waterloo, Comp2 to occur Jan 2025)
  - Alex Fedorov (PhD, IST Austria, proposal 2024 [defense to occur])
  - Zhuanhao Wu (PhD, University of Waterloo, proposal 2024 [defense to occur])
  - Fadhil Abubaker (MMath, University of Waterloo, 2024)
  - Yaodong Sheng (MSc, Lehigh University, 2024)
  - Ahmed Fahmy (PhD, University of Waterloo, 2024)
  - Colby Alexander Parsons (MMath, University of Waterloo, 2023)
  - Bryant Curto (MMath, University of Waterloo, 2022)
  - Thierry Delisle (PhD, University of Waterloo, 2022)
  - Mubeen Zulfiqar (MMath, University of Waterloo, 2022)
  - Sakib Chowdhury (MMath, University of Waterloo, 2021)
  - Basil Alkhatib (MMath, University of Waterloo, 2021)
  - Anastasia Postnikova (BSc, ITMO University, 2021)
  - Bryce Sandlund (PhD, University of Waterloo, 2021)
  - Diego Cepada (MMath, University of Waterloo, 2020)
  - Nan Li (MMath, University of Waterloo, 2020)
  - Mehrdad Giv (MSc, Calgary, 2020)

Service

- **Committees**
  - Graduate recruiting committee (Waterloo, F2024-S2025)
  - Awards committee (Waterloo, F2024-S2025)
  - School advisory committee on faculty appointments (Waterloo, F2023-S2024)
  - Graduate recruiting committee (Waterloo, F2022-W23)
  - School advisory committee on faculty appointments (Waterloo, F2021-S2022)
  - Graduate recruiting committee (Waterloo, F2020-S2021)
  - School advisory committee on faculty appointments (Waterloo, 2019)
  - Graduate recruiting committee (Waterloo, 2018)
  - CS Chair search committee (Toronto, 2015)
- **Grant reviewer** for the Israel Science Foundation, 2023
- **Discussion panel member** for
  - Graduate recruiting student panel (University of Waterloo, Apr 2021)
  - Graduate skills seminars (University of Waterloo, Sep 2020)
  - UROC undergraduate research conference (University of Waterloo, Sep 2019)
- **Student reference letters for graduate school**
  - three students in fall 2023
  - three students in fall 2022
  - three students in fall 2021
- **Tenure recommendation letters (2x)**, York University (2010)

Professional Activities

- **Association for Computing Machinery – Lifetime member**
- **Program committee member** for
  - PPoPP'25
  - PPoPP'24 (10 papers)
  - SPAA'22 (10 papers)
  - SIROCCO'22 (5 papers)
  - ICDCS'21 (2 papers) [duties curtailed because of hospitalization]
  - PPoPP'21 (10 papers)
  - PODC'21 (17 papers)
  - PODC'20 (19 papers)
  - PODC'19 (23 papers)
  - PPoPP'19 (extended review committee)
  - ICDCS'18
  - Additionally was invited to PCs for PPoPP'25, FCPC'24, SPAA'24, DISC'24, PODC'24, ICDCS'23, SPAA'23, DISC'22, PODC'22, OPODIS'22, SPAA'21, SPAA'20 (but declined due to timing)
- **Publication chair** for PPoPP'19.
  A rough guide that I wrote for subsequent publication chairs (currently in use).
- **Artifact evaluation committee member** for PPoPP'16.
- **Conference paper reviews** for PODC'11, DISC'12, PODC'13, DISC'14, PODC'14, PPoPP'16, SIROCCO'16, SPAA'16, DISC'16, SPAA'17, RANDOM'17, DISC'17, PODC'18 (4x), DISC'18 (3x), SPAA'19, ESA'19 (2x), DISC'19 (2x), APLAS'20, DISC'20, DISC'22, PODC'23, DISC'24.
- **Journal reviews** for the Journal of the ACM (JACM) [1x 2023], the ACM Journal of Distributed Computing (DIST) [5x], the IEEE Journal of Transactions on Parallel and Distributed Systems (TPDS) [3x], IEEE Transactions on Parallel Computing (TOPC) [1x 2025], and the Elsevier Journal of Logical and Algebraic Methods in Programming (JLAMP) [2x].
- **Grant reviewer** for NSERC (2025) and the Israeli Science Foundation (2023).