Reading list for CS 260: Algorithm Engineering

YAN GU, University of California, Riverside

The following is the Reading list for CS 260: Algorithm Engineering. You should read and turn in paper readings from the this list of papers.

1 GENERAL IDEAS FOR PARALLEL ALGORITHMS

Optimal Parallel Algorithms in the Binary-Forking Model [12]
Parallel In-Place Algorithms: Theory and Practice [29]
Internally Deterministic Parallel Algorithms Can Be Fast [11]

2 SEQUENCE ALGORITHMS

Low Depth Cache-Oblivious Algorithms [15]
Prefix Sums and Their Applications [7]
In-place Parallel Super Scalar Samplesort (IPS4o) [4]
A Top-Down Parallel Semisort [31]
Sequential Random Permutation, List Contraction and Tree Contraction are Highly Parallel [39]
The input/output complexity of sorting and related problems [2]

3 DATA STRUCTURES

Just Join for Parallel Ordered Sets [8]
PAM: Parallel Augmented Maps [41]
Phase-concurrent Hash Tables for Determinism [37]
On Supporting Efficient Snapshot Isolation for Hybrid Workloads with Multi-Versioned Indexes [43]
A Simple Parallel Cartesian Tree Algorithm and its Application to Parallel Suffix Tree Construction [21]

4 GRAPHS

Ligra: A Lightweight Graph Processing Framework for Shared Memory [36]
Smaller and Faster: Parallel Processing of Compressed Graphs with Ligra+ [38]
Julienne: A Framework for Parallel Graph Algorithms using Work-efficient Bucketing [23]
Sage: Semi-Asymmetric Parallel Graph Algorithms for NVRAMs [27]
GraphChi: Large-Scale Graph Computation on Just a PC [33]
Work-Efficient Parallel Union-Find [40]
STINGER: High Performance Data Structure for Streaming Graphs [28]
Low-Latency Graph Streaming Using Compressed Purely-Functional Trees [25]
Parallel Shortest-Paths Using Radius Stepping [19]
Implicit Decomposition for Write-Efficient Connectivity Algorithms [6]

Author's address: Yan Gu, University of California, Riverside.

Optimizing Ordered Graph Algorithms with GraphIt [46]
Theoretically Efficient Parallel Graph Algorithms Can Be Fast and Scalable [24]
Parallel graph decompositions using random shifts [35]
Delta-stepping: a parallelizable shortest path algorithm [34]

5 GEOMETRY PROCESSING

Parallelism in Randomized Incremental Algorithms [16]
Fast, Parallel, and Asynchronous Construction of BVHs for Ray Tracing Animated Scenes [44]
Efficient BVH Construction via Approximate Agglomerative Clustering [30]
Parallel Range, Segment and Rectangle Queries with Augmented Maps [42]
Theoretically-Efficient and Practical Parallel DBSCAN [45]
Theory and Practice of Parallel Euclidean Minimum Spanning Tree and Hierarchical Spatial Clustering
Design and Implementation of a Practical Parallel Delaunay Algorithm [20]
Randomized Incremental Convex Hull is Highly Parallel [18]
Parallel Write-Efficient Algorithms and Data Structures for Computational Geometry [17]

6 INCREMENTAL CONSTRUCTION

Internally Deterministic Parallel Algorithms Can Be Fast [11]
Parallelism in Randomized Incremental Algorithms [16]
Sequential Random Permutation, List Contraction and Tree Contraction are Highly Parallel [39]
Greedy Sequential Maximal Independent Set and Matching are Parallel on Average [13]
Randomized Incremental Convex Hull is Highly Parallel [18]

7 SCHEDULING

Scheduling multithreaded computations by work stealing [22]
Thread Scheduling for Multiprogrammed Multiprocessors [3]
The Data Locality of Work Stealing [1]

8 WRITE-EFFICIENT ALGORITHMS

Sorting with Asymmetric Read and Write Costs [9]
Efficient Algorithms with Asymmetric Read and Write Costs [10]
Parallel Algorithms with Asymmetric Read and Write Costs [5]
Implicit Decomposition for Write-Efficient Connectivity Algorithms [6]
The Parallel Persistent Memory Model [14]
Parallel Write-Efficient Algorithms and Data Structures for Computational Geometry [17]
Algorithmic Building Blocks for Asymmetric Memories [32]
Sage: Parallel Semi-Asymmetric Graph Algorithms for NVRAMs [26]

REFERENCES

[1] Umut A. Acar, Guy E. Blelloch, and Robert D. Blumofe. 2002. The Data Locality of Work Stealing. *Theoretical Computer Science (TCS)* 35, 3 (2002).

[2] Alok Aggarwal and Jeffrey S. Vitter. 1988. The Input/Output Complexity of Sorting and Related Problems. *Commun. ACM* 31, 9 (1988).

[3] N. S. Arora, R. D. Blumofe, and C. G. Plaxton. 2001. Thread Scheduling for Multiprogrammed Multiprocessors. *Theory of Computing Systems (TOCS)* 34, 2 (01 Apr 2001).

[4] Michael Axtmann, Sascha Witt, Daniel Ferizovic, and Peter Sanders. 2017. In-place parallel super scalar samplesort (ipsssso). In *European Symposium on Algorithms (ESA)*.

[5] Naama Ben-David, Guy E. Blelloch, Jeremy T. Fineman, Phillip B. Gibbons, Yan Gu, Charles McGuffey, and Julian Shun. 2016. Parallel Algorithms for Asymmetric Read-Write Costs. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[6] Naama Ben-David, Guy E. Blelloch, Jeremy T Fineman, Phillip B Gibbons, Yan Gu, Charles McGuffey, and Julian Shun. 2018. Implicit Decomposition for Write-Efficient Connectivity Algorithms. In *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*.

[7] Guy E. Blelloch. 1993. Prefix Sums and Their Applications. In *Synthesis of Parallel Algorithms*, John Reif (Ed.). Morgan Kaufmann.

[8] Guy E. Blelloch, Daniel Ferizovic, and Yihan Sun. 2016. Just Join for Parallel Ordered Sets. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[9] Guy E. Blelloch, Jeremy T. Fineman, Phillip B. Gibbons, Yan Gu, and Julian Shun. 2015. Sorting with Asymmetric Read and Write Costs. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[10] Guy E. Blelloch, Jeremy T. Fineman, Phillip B. Gibbons, Yan Gu, and Julian Shun. 2016. Efficient Algorithms with Asymmetric Read and Write Costs. In *European Symposium on Algorithms (ESA)*.

[11] Guy E. Blelloch, Jeremy T. Fineman, Phillip B. Gibbons, and Julian Shun. 2012. Internally deterministic parallel algorithms can be fast. In *ACM Symposium on Principles and Practice of Parallel Programming (PPOPP)*.

[12] Guy E. Blelloch, Jeremy T. Fineman, Yan Gu, and Yihan Sun. 2020. Optimal parallel algorithms in the binary-forking model. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[13] Guy E. Blelloch, Jeremy T Fineman, and Julian Shun. 2012. Greedy sequential maximal independent set and matching are parallel on average. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[14] Guy E. Blelloch, Phillip B Gibbons, Yan Gu, Charles McGuffey, and Julian Shun. 2018. The Parallel Persistent Memory Model. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[15] Guy E. Blelloch, Phillip B. Gibbons, and Harsha Vardhan Simhadri. 2010. Low depth cache-oblivious algorithms. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[16] Guy E. Blelloch, Yan Gu, Julian Shun, and Yihan Sun. 2016. Parallelism in Randomized Incremental Algorithms. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[17] Guy E. Blelloch, Yan Gu, Julian Shun, and Yihan Sun. 2018. Parallel Write-Efficient Algorithms and Data Structures for Computational Geometry. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[18] Guy E. Blelloch, Yan Gu, Julian Shun, and Yihan Sun. 2020. Randomized Incremental Convex Hull is Highly Parallel. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[19] Guy E. Blelloch, Yan Gu, Yihan Sun, and Kanat Tangwongsan. 2016. Parallel Shortest Paths Using Radius Stepping. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[20] Guy E Blelloch, Gary L Miller, Jonathan C Hardwick, and Dafna Talmor. 1999. Design and implementation of a practical parallel Delaunay algorithm. *Algorithmica* 24, 3-4 (1999), 243–269.

[21] G. E. Blelloch and J. Shun. 2011. A Simple Parallel Cartesian Tree Algorithm and its Application to Suffix Tree Construction. In *Algorithm Engineering and Experiments (ALENEX)*.

[22] Robert D. Blumofe and Charles E. Leiserson. 1998. Space-Efficient Scheduling of Multithreaded Computations. *SIAM J. Scientific Computing* 27, 1 (1998).

[23] Laxman Dhulipala, Guy E. Blelloch, and Julian Shun. 2017. Julienne: A Framework for Parallel Graph Algorithms using Work-efficient Bucketing. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[24] Laxman Dhulipala, Guy E. Blelloch, and Julian Shun. 2018. Theoretically Efficient Parallel Graph Algorithms Can Be Fast and Scalable. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[25] Laxman Dhulipala, Guy E Blelloch, and Julian Shun. 2019. Low-latency graph streaming using compressed purely-functional trees. In *ACM Conference on Programming Language Design and Implementation (PLDI)*. 918–934.

[26] Laxman Dhulipala, Charlie McGuffey, Hongbo Kang, Yan Gu, Guy E Blelloch, Phillip B Gibbons, and Julian Shun. 2019. Semi-Asymmetric Parallel Graph Algorithms for NVRAMs. *arXiv preprint arXiv:1910.12310* (2019).

[27] Laxman Dhulipala, Charlie McGuffey, Hongbo Kang, Yan Gu, Guy E Blelloch, Phillip B Gibbons, and Julian Shun. 2020. Semi-Asymmetric Parallel Graph Algorithms for NVRAMs. *Proceedings of the VLDB Endowment (PVLDB)* 13, 9 (2020).

[28] David Ediger, Robert McColl, Jason Riedy, and David A Bader. 2012. Stinger: High performance data structure for streaming graphs. In *IEEE Conference on High Performance Extreme Computing (HPEC)*. 1–5.

[29] Omer Gu, Yan Obeya and Julian Shun. 2020. Parallel In-Place Algorithms: Theory and Practice.

[30] Yan Gu, Yong He, Kayvon Fatahalian, and Guy Blelloch. 2013. Efficient BVH construction via approximate agglomerative clustering. In *High-Performance Graphics (HPG)*.

[31] Yan Gu, Julian Shun, Yihan Sun, and Guy E Blelloch. 2015. A top-down parallel semisort. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[32] Yan Gu, Yihan Sun, and Guy E. Blelloch. 2018. Algorithmic Building Blocks for Asymmetric Memories. In *European Symposium on Algorithms (ESA)*.

[33] Aapo Kyrola, Guy E. Blelloch, and Carlos Guestrin. 2012. GraphChi: Large-Scale Graph Computation on Just a PC. In *USENIX conference on Operating Systems Design and Implementation (OSDI)*.

[34] Ulrich Meyer and Peter Sanders. 2003. Δ-stepping: a parallelizable shortest path algorithm. *Journal of Algorithms* 49, 1 (2003), 114–152.

[35] Gary L Miller, Richard Peng, and Shen Chen Xu. 2013. Parallel graph decompositions using random shifts. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[36] Julian Shun and Guy E. Blelloch. 2013. Ligra: A Lightweight Graph Processing Framework for Shared Memory. In *ACM Symposium on Principles and Practice of Parallel Programming (PPOPP)*.

[37] Julian Shun and Guy E Blelloch. 2014. Phase-concurrent hash tables for determinism. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*.

[38] Julian Shun, Laxman Dhulipala, and Guy E Blelloch. 2015. Smaller and faster: Parallel processing of compressed graphs with Ligra+. In *IEEE Data Compression Conference (DCC)*. 403–412.

[39] Julian Shun, Yan Gu, Guy E. Blelloch, Jeremy T Fineman, and Phillip B Gibbons. 2015. Sequential random permutation, list contraction and tree contraction are highly parallel. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 431–448.

[40] Natcha Simsiri, Kanat Tangwongsan, Srikanta Tirthapura, and Kun-Lung Wu. 2016. Work-efficient parallel union-find with applications to incremental graph connectivity. In *European Conference on Parallel Processing (Euro-Par)*.

[41] Yihan Sun and Guy E Blelloch. 2018. Parallel Range and Segment Queries with Augmented Maps. *arXiv preprint:1803.08621* (2018).

[42] Yihan Sun and Guy E Blelloch. 2019. Parallel Range, Segment and Rectangle Queries with Augmented Maps. In *SIAM Symposium on Algorithm Engineering and Experiments (ALENEX)*. 159–173.

[43] Yihan Sun, Guy E Blelloch, Wan Shen Lim, and Andrew Pavlo. 2019. On supporting efficient snapshot isolation for hybrid workloads with multi-versioned indexes. *Proceedings of the VLDB Endowment (PVLDB)* 13, 2 (2019), 211–225.

[44] Ingo Wald, Thiago Ize, and Steven G Parker. 2008. Fast, parallel, and asynchronous construction of BVHs for ray tracing animated scenes. *Computers & Graphics* 32, 1 (2008), 3–13.

[45] Yiqiu Wang, Yan Gu, and Julian Shun. 2020. Theoretically-Efficient and Practical Parallel DBSCAN. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*. 2555–2571.

[46] Yunming Zhang, Ajay Brahmakshatriya, Xinyi Chen, Laxman Dhulipala, Shoaib Kamil, Saman Amarasinghe, and Julian Shun. 2020. Optimizing ordered graph algorithms with GraphIt. In *ACM/IEEE International Symposium on Code Generation and Optimization (CGO)*. 158–170.