A Survey on Counting Networks

C. BUSCH
Brown University, Rhode Island, USA

M. HERLIHY
Brown University, Rhode Island, USA

Abstract

In the *counting problem*, asynchronous concurrent processes repeatedly assign themselves successive values, such as integers or locations in memory. *Counting networks* are a class of concurrent data structures that support highly concurrent counting in a way that minimizes serialization and memory contention. In the past six years, counting networks have been the focus of a growing body of research. This paper surveys recent work on counting networks.

Keywords: Counting Network, Balancing Network, Balancer, Contention.

1 Introduction

Many fundamental multi-processor coordination problems can be expressed as *counting problems:* processors collectively assign successive values from a given range, such as addresses in memory or destinations in an interconnection network. Aspnes, Herlihy, and Shavit [5] offered a new approach to solving such problems by introducing *counting networks*, a class of networks that can be used to count.

Counting networks, like sorting networks [6, 4, 11], are constructed from simple two-input, two-output computing elements called *balancers*, connected to one another by wires. A process accesses the counting network by issuing tokens. A counting network counts any number of input tokens even if they arrive at different times. The tokens may be distributed unevenly among the input wires, and may propagate through the network asynchronously. Counting networks achieve a high level of throughput by decomposing interactions among processes into pieces that can be performed in parallel. This decomposition has two performance benefits: it eliminates serial bottlenecks and it reduces memory contention. Counting networks are also non-blocking: processes that undergo halting failures or delays while using a counting network do not prevent other processes from making progress.

A *balancer* is a switch with 2 input wires and 2 output wires, numbered 0 and 1. Tokens enter the balancer on input wires, and exit on output wires. The i-th token to enter the balancer leaves on output wire $i \bmod 2$. A *balancing network* is constructed by wiring balancers together in an acyclic network. A network of *width* w has w input wires and w output wires. Tokens enter the network on input wires, propagate through the balancers, and leave on the output wires. The *depth* of the network is the maximum number of balancers traversed by any token. A balancing network is in a *quiescent state* if no tokens are in the network, that is all the tokens that have entered the network have exited it. We denote by y_i the number of tokens that exit from the i-th output wire of a network.

A *counting network* of width w is a balancing network whose outputs $y_0, \ldots, y_{w-1}$ satisfy the following *step property*:

In any quiescent state, $0 \leq y_i - y_j \leq 1$ for any $i < j$.

A counting network can easily be adapted to count the number of tokens that have entered the network simply by adding a "local counter" to each output wire i, so that tokens coming out of that wire are consecutively assigned the numbers $i, i + w, i + 2w, \ldots, i + (y_i - 1)w$. Applications of counting networks include shared pools and stacks, load balancing, and barriers [5, 17, 23].

An important measurement of the efficiency of a counting network is its depth, and this is because it is equal to the number of the memory locations that a process must access before its increment request is satisfied. Naturally, minimizing counting network depth remains a goal of much counting network research.

Aspnes, Herlihy, and Shavit [5] give two distinct counting network constructions, isomorphic to the Bitonic sorting network of Batcher [6] and to the Periodic Balanced sorting network of Dowd, Perl, Rudolph and Saks [11]. Each of these networks has depth less than or equal to $\log^2 w$, using $(w \log^2 w)/2$ balancers or fewer.

2 Topics in Counting Networks

2.1 Impossibility Results

A balancing network and a comparison network are *isomorphic* if one can be constructed from the other by replacing balancers by comparators or vice versa. Aspnes, Herlihy, and Shavit [5] show that every counting network is isomorphic to a sorting network, implying that the $\Omega(\log w)$ lower bound for sorting network depth holds also for counting networks.

Aharonson and Attiya [1] generalized the notion of a balancer to have different number of input and output wires. Specifically, a balancer can have k input wires and l output wires for any $k, l \geq 1$, where k and l are not necessarily equal (for a regular balancer we have $k = l = 2$). Using such "irregular" balancers we can construct counting networks with different number of input and output wires (see Section 2.6). They showed that the output width of a counting network must be such that its prime factors must divide the output widths of the balancers that it uses. Therefore, using only regular balancers with input and output width 2 one can construct counting networks only of width 2^k. To construct counting networks of different widths one must use balancers with p outputs, where $p \geq 2$ (see Section 2.5).

Busch and Mavronicolas [9] used combinatorial properties of balancing networks to show that the number P^d/w must be an integer, where P is the least common multiple of the different widths of balancers in the network, and d and w are the depth and width of the network, respectively. This result generalizes the result of Aharonson and Attiya [1] and together gives another proof of the lower bound on the depth.

2.2 Small-Depth Constructions

We list the counting network constructions of smallest known depth. Klugerman [18] presents a simple explicit counting network construction with width w and depth $O((\log w) \log \log w)$. Klugerman and Plaxton [20] present an explicit counting network construction with depth $O(c^{\log^* w} \log w)$ (for some positive constant c). They also present a randomized counting network construction with depth $O(\log w)$, that counts with extremely high probability. Using this construction they prove the existence of a counting network of depth $O(\log w)$, that matches the lower bound $\Omega(\log w)$. Klugerman [19] extends this result and presents a polynomial-time method to construct an $O(\log w)$ depth counting network. All the above constructions use as a building block the AKS sorting network [4] whose depth expression $O(\log w)$ hides huge constants and subsequently all these networks are impractical.

2.3 Linearizability

Herlihy, Shavit, and Waarts [16] defined the class of *linearizable counting networks*. These networks assure that the order of the values returned by the network reflect the real-time order in which they were requested. Specifically, if a token a exits the counting network before token b enters it then the value of a is smaller than the value of b. In the same paper a lower bound of $O(w)$ was established for the depth of any linearizable counting network of width w. Furthermore, two constructions were presented, the first with depth $O(w)$, where a token takes an average of $O(w)$ steps and an individual token may take an infinite number of steps if it is infinitely overtaken, and the second construction with depth $O(w^2)$ where a token takes at most $O(w^2)$ steps.

Lynch, Shavit, Shvartsman, and Touitou [21] show that under certain timing conditions the *uniform* non-linearizable counting networks exhibit linearizable behavior. In a uniform network the paths from its inputs to its outputs that go through a balancer all have the same length. The bitonic and periodic counting networks [5] are examples of uniform networks that both have depth $O(\log^2 w)$. They show that these networks are linearizable if the time it takes a slow token to traverse a wire or balancer is no more than twice than that of a fast token. Mavronicolas, Papatriantafilou, and Tsigas [22] extend the analysis of timing conditions for linearizability and provide tighter bounds for such conditions.

2.4 Randomized Balancers

Aiello, Venkatesan, and Yung [3] introduce the *randomized balancer*. A 2-output randomized balancer sends its first output token to a randomly chosen output wire and the second on the opposite wire, and similarly for the rest of the tokens. Like the deterministic balancer, the difference between the tokens on the output wires is at most 1, but, the wire with the excess number of tokens is random (instead of the first one at the deterministic balancer). They present an explicit construction of an $O(\log w)$ depth randomized counting network of width w from randomized balancers and deterministic balancers, that counts with high probability. The randomized balancers appear only for a logarithmic number of layers in the beginning of the construction and then the AKS sorting network [4] is used as a subroutine.

2.5 Arbitrary Widths

We list the counting network constructions that use balancers of the same input and output width, bigger than 2, and yield network widths that are not a power of 2 (see Section 2.1). Aharonson and Attiya [1] constructed a counting network of width $w = p2^k$ and depth $O(\log^3(w/p))$ from balancers of width 2 and p. They also construct networks of arbitrary width by taking a standard counting network and linking the excess output wires to the excess input wires, resulting in a cyclic network. Busch, Hardavellas, and Mavronicolas [7] give a construction of width $w = p2^k$ and depth $O(\log^2(w/p))$ using balancers of width 2 and p. Felten, LaMarca, and Ladner [13] give a construction of width $w = 2^k$ and depth $O(\log^2 w)$ from balancers of width 2^ℓ, as well as a construction of width $w = p2^k$ using also balancers of width p.

Klugerman [19] gives a counting network construction of arbitrary width w and depth $O((\log w) \log\log w)$ from balancers of width p, where p ranges over the prime factors of w. This construction is based on the AKS sorting network [4], and is therefore not practical. Busch and Herlihy [8] give a practical construction of arbitrary width w and depth $O(\log^2 w)$ from balancers of width at most the maximum factor of w, where w can be factored in any way (including the prime factorization).

2.6 Irregular Constructions

We list the counting network constructions that use "irregular" balancers, whose output widths are different than the input widths, and yield counting networks with different input and output widths. Shavit and Zemach [24], as part of a more general result, present the *counting tree* constructed from 1-input, 2-output balancers. This counting network has the form of a binary tree with input width 1, output width w and depth $log w$.

Aiello, Venkatesan, and Yung [3] present a counting network construction of input width w output width $w \log w$ and optimal depth $O(\log w)$, using 1-input, 2-output balancers and regular balancers of width 2. This construction uses as a subroutine the AKS network [4] and is therefore not practical. Busch and Mavronicolas [10] present a counting network construction with input width $w = 2^k$ output width $p2^l$ and depth $O(\log^2 w)$, using 2-input, p-output balancers and regular balancers of width 2.

2.7 Supporting Decrement

Counting networks have been used to support increment operations, where a token represents the request for an increment of 1. Shavit and Touitou [23] showed that the class of counting networks that have the shape of a binary tree can support decrement operations, together with the increment operations. They did so by introducing a new type of token for the decrement operation named *antitoken* that represents the request for an increment of -1. Aiello, Busch, Herlihy, Mavronicolas, Shavit, and Touitou [2] generalized this result and showed that *any* counting network can support the decrement operation (including networks that use arbitrary width balancers and irregular balancers, see Sections 2.1, 2.5, and 2.6).

2.8 Contention Analysis

Dwork, Herlihy, and Waarts [12] introduce a formal model for measuring *contention*, the extent to which concurrent processors access the same memory location simultaneously. For counting networks they define the *amortized contention* which measures the contention in the worst-case and in the limit when many processors access the counting network concurrently. This measures the number of "stall" steps that a token is charged each time another token bypasses it in a balancer. They found that the amortized contention in the bitonic and periodic [5] counting networks are $\Theta(n \log^2 w/w)$ and $\Theta(n \log^3 w/w)$, respectively. Hardavellas, Damianos, and Mavronicolas [14] improve on the bound of the periodic network and find that its amortized contention is $\Theta(n \log^2 w/w)$.

2.9 Experimental Results

Herlihy, Lim, and Shavit [15] describe experiments investigating the scalability of a variety of counting techniques for large-scale multiprocessors. They compare counting techniques based on: (1) spin locks, (2) message passing, (3) distributed queues, (4) software combining trees, and (5) counting networks. The comparison is based on a set of simple benchmarks on a simulated 64-processor MIT Alewife Machine. They observed that centralized locking techniques do not perform well on scalable architectures such as Alewife, because serialization limits achievable throughput, and contention degrades performance. Moreover, they observed that both counting networks and combining trees substantially outperform the other methods, because they avoid serialization and contention, although combining tree throughput is more sensitive to load variations than counting networks. Finally, a comparison of shared-memory and message-passing implementations of counting networks and combining trees shows that message-passing implementations have substantially higher throughput.

3 Conclusions

We believe that the work surveyed here represents a start toward a general theory of low-contention data structures. Work is needed to develop other primitives, to derive upper and lower bounds and new performance measures. For example, smoothing networks, balancing networks that smooth but do not necessarily count, are interesting in their own right since they can be used for problems such as load balancing. Concurrent data structures that support other kinds of objects, such as concurrent priority queues, remain poorly understood. We believe that counting networks and their relatives deserve further study.

References

[1] AHARONSON, E., AND ATTIYA, H. Counting networks with arbitrary fan-out. *Distributed Computing 8*, 4 (1995), 163–169.

[2] AIELLO, W., BUSCH, C., HERLIHY, M., MAVRONICOLAS, M., SHAVIT, N., AND TOUITOU, D. Supporting increment and decrement operations in balancing networks. In *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS'99)* (Trier, Germany, 1999). To appear.

[3] AIELLO, W., VENKATESAN, R., AND YUNG, M. Coins, weights and contention in balancing networks. In *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing (PODC'94)* (Los Angeles, Aug. 1994), pp. 193–205.

[4] AJTAI, M., KOMLÓS, J., AND SZEMERÉDI, E. An $O(n \log n)$ sorting network. *Combinatorica 3* (1983), 1–19.

[5] ASPNES, J., HERLIHY, M., AND SHAVIT, N. Counting networks. *Journal of the ACM 41*, 5 (Sept. 1994), 1020–1048.

[6] BATCHER, K. E. Sorting networks and their applications. In *Proc. AFIPS 1968 Spring Jt. Computer Conf.* (Washington, D.C., 1968), vol. 32, Thompson Book,Co., pp. 307–314.

[7] BUSCH, C., HARDAVELLAS, N., AND MAVRONICOLAS, M. Contention in counting networks (abstract). In *Proceedings of the 13th annual ACM Symposium on Principles of Distributed Computing (PODC'94)* (Los Angeles, Aug. 1994), p. 404.

[8] BUSCH, C., AND HERLIHY, M. Small-depth counting networks of arbitrary width. Submitted to the 18th annual ACM Symposium on Principles of Distributed Computing (PODC'99), Nov. 1998.

[9] BUSCH, C., AND MAVRONICOLAS, M. A combinatorial treatment of balancing networks. *Journal of the ACM 43*, 5 (Sept. 1996), 794–839.

[10] BUSCH, C., AND MAVRONICOLAS, M. An efficient counting network. In *Proceedings of the 1st Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing (IPPS/SPDP'98)* (Mar. 1998), pp. 380–385.

[11] DOWD, M., PERL, Y., RUDOLPH, L., AND SAKS, M. The periodic balanced sorting network. *Journal of the ACM 36*, 4 (Oct. 1989), 738–757.

[12] DWORK, C., HERLIHY, M., AND WAARTS, O. Contention in shared memory algorithms. *Journal of the ACM 44*, 6 (Nov. 1997), 779–805.

[13] FELTEN, E. W., LAMARCA, A., AND LADNER, R. Building counting networks from larger balancers. Tech. Rep. TR 93-04-09, University of Washington, Apr. 1993.

[14] HARDAVELLAS, N., KARAKOS, D., AND MAVRONICOLAS, M. Notes on sorting and counting networks. In *Proceedings of the 7th International Workshop on Distributed Algorithms (WDAG'93)* (Lausanne, Switzerland, Sept. 1993), vol. 725 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 234–248.

[15] HERLIHY, M., LIM, B.-H., AND SHAVIT, N. Scalable concurrent counting. *ACM Transactions on Computer Systems 13*, 4 (Nov. 1995), 343–364.

[16] HERLIHY, M., SHAVIT, N., AND WAARTS, O. Linearizable counting networks. *Distributed Computing 9*, 4 (1996), 193–203.

[17] KAPIDAKIS, S., AND MAVRONICOLAS, M. Distributed, low contention task allocation. In *Proceedings of the 8th IEEE Symposium on Parallel and Distributed Processing (SPDP'96)* (Washington, Oct. 1996), pp. 358–365.

[18] KLUGERMAN, M. Lecture 17: Counting networks. In *Research Seminar Series 15: Advanced Parallel and VLSI Computation*. MIT Press, 1991, pp. 153–161.

[19] KLUGERMAN, M. *Small-Depth Counting Networks and Related Topics*. PhD thesis, Department of Mathematics, Massachusetts Institute of Technology, Sept. 1994.

[20] KLUGERMAN, M., AND PLAXTON, C. G. Small-depth counting networks. In *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing (STOC'92)* (Victoria, B.C., Canada, May 1992), pp. 417–428.

[21] LYNCH, N., SHAVIT, N., SHVARTSMAN, A., AND TOUITOU, D. Counting networks are practically linearizable. In *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing (PODC'96)* (New York, May 1996), pp. 280–289.

[22] MAVRONICOLAS, M., PAPATRIANTAFILOU, M., AND TSIGAS, P. The impact of timing on linearizability in counting networks. In *Proceedings of the 11th International Parallel Processing Symposium (IPPS'97)* (Los Alamitos, Apr. 1997), pp. 684–688.

[23] SHAVIT, N., AND TOUITOU, D. Elimination trees and the construction of pools and stacks. *Theory of Computing Systems 30*, 6 (Nov./Dec. 1997), 545–570.

[24] SHAVIT, N., AND ZEMACH, A. Diffracting trees. *ACM Transactions on Computer Systems 14*, 4 (Nov. 1996), 385–428.