An Efficient Counting Network

Costas Busch
Department of Computer Science
Brown University
Providence, RI 02912
cb@cs.brown.edu

Marios Mavronicolas
Department of Computer Science
University of Cyprus
Nicosia 1678, Cyprus
mavronic@turing.cs.ucy.ac.cy

Abstract

Counting networks were introduced as a new class of concurrent, distributed, low contention data structures suitable for implementing shared counters. Their structure is similar to that of sorting networks. High-performance asynchronous multiprocessing requires counting networks to both have small depth *and incur low* contention. *In order to achieve this, we relax in this work the requirement that the input width of the counting network is equal to its output width. More specifically, we present an explicit, deterministic construction of a counting network with t input width and w output width, where $t \leq w$, $t = 2^k$ and $w = p2^l$. This construction is practical and achieves depth $O(\lg^2 t)$ which is independent from the output width w. Furthermore, by taking w to be $\Theta(t \lg t)$ it incurs an amortized contention of the order $O((n \lg t)/t)$, where n is the concurrency, which improves by a logarithmic factor over all previously known practical counting networks constructions of width t.*

1. Introduction

A shared counter can be easily implemented using a single shared *Fetch&Increment* variable. However, empirically, the time to access a shared variable grows at least linearly with the *contention*, the extent to which concurrent processors simultaneously attempt to access the variable. Aspnes *et al.* [3] suggested the *counting network* as an alternative approach for implementing shared counters.

Counting networks are constructed from simple elements called *balancers* in a similar way that sorting networks are constructed from comparators (see [4, 10]). Loosely speaking, a balancer can be thought of as a toggle mechanism with p input and output wires that receives tokens from its input wires and forwards them to its output wires (see [3, 5, 8]). When a token appears on an input wire, it takes the output wire to which the toggle is set, and toggles the balancer so that the input next to come will leave on the next output wire. If the toggle was set to the last output wire it is set back to the first output wire.

One can connect a collection of balancers to form a *balancing network*. This is done by connecting output wires from some balancers to input wires of others. The remaining unconnected input and output wires are the input and output wires, respectively, of the network. The number of input and output wires is the same and is called the *width* t of the network. Like the balancer, the balancing network receives tokens in its inputs and forwards them in its outputs. A counting network is a balancing network that has the *step property* (see Section 2), a property which makes it able to behave like a counter. A processor that wants to obtain a new value from the counter traverses the network by issuing a token, and according to the output it leaves from the network it takes an appropriate value.

In this work we deviate from the "traditional" approach and we construct counting networks which have different input and output widths (different number of input and output wires). In our construction the input width t is smaller or equal to the output width w. More specifically, we have $t = 2^k$, $w = p2^l$ and $k \leq l$. Our counting network, denoted as $C(t, w)$, is constructed from regular balancers with 2 inputs and outputs and from balancers with different number of input and output wires. A q-input, p-output balancer behaves in the same as a regular balancer, that is, a token is received from one of its q inputs and is forwarded to one of its p outputs using the same toggle mechanism with p settings (see also [2, 11]). In $C(t, w)$ we use balancers with $q = 2$ and $p \geq 2$. In figure 1 we see the construction of $C(4, 8)$, where the balancers are drawn with vertical lines and the wires are drawn with horizontal lines. We see that this construction uses 2-input, 2-output balancers, and 2-input, 4-output balancers.

Our construction improves over all known practical constructions in terms of *depth* and *contention*, two important measures for balancing and counting networks. The depth of a balancing network is the maximal path length from an

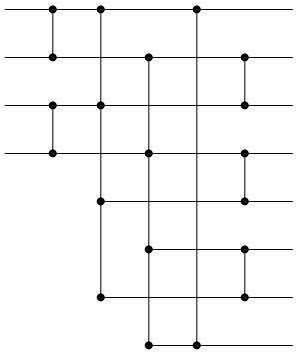

Figure 1. The counting network $C(4,8)$.

input wire to an output wire. The depth is important since the number of memory locations that a processor may have to access, before its incremental request is satisfied, is at most the depth of the network. The contention is the extent to which concurrent processors access the same memory location (the balancer in our case) at the same time. The *amortized contention*, defined by Dwork *et al.* [7], measures contention in the worst-case and in the limit when many processors access the balancing network concurrently. In order to achieve good performance in a counting network it is necessary to achieve both small depth and low contention (see [6]).

The traditional practical counting networks known so far achieve depth $O(\lg^2 t)$, where t is the width of the network. Such networks are the bitonic and periodic counting networks [3], which use 2-input, 2-output balancers, and other constructions which use balancers of larger widths [1, 8, 9]. The amortized contention that is achieved by these networks is of the order $O((n \lg^2 t)/t)$, where n is the processor concurrency. It is easy to see that in these networks if we need low contention we have to use large widths. However, this has the side effect that it increases the depth of the network. Therefore, there exists a trade off between the choice of the appropriate depth and contention which is related to the width of the counting network.

Our construction, due to its irregular structure, achieves depth $O(\lg^2 t)$ which is independent from its output width. Simultaneously, the amortized contention is $\Theta((n \lg^2 t)/w + (n \lg t)/t)))$ which means that by increasing the output width the contention drops. Therefore, for any fixed input width t we can decrease the contention by increasing only the output width, while preserving the depth of the network. This way, we avoid the trade off between depth and contention found in traditional networks. Actually, by making w to be of the order $\Theta(t \lg t)$ we achieve amortized contention of the order $O((n \lg t)/t)$ which improves by a logarithmic factor over all known best practical counting network constructions.

The performance of our network can be explained as follows. A balancing network can be divided in layers, where each layer contains the balancers that are in a specific depth. In the traditional counting networks the width of all the layers is equal to t, making the contention to be the same for each layer. On the other hand, in our network, only the first $\lg t$ layers have width t, and the rest layers, which are the majority, have width w. By taking w to be larger than t more balancers are available at each of the last layers. Thus the contention of the balancers of these layers decreases as w increases, making the total contention to decrease.

The construction uses as a building block a network with a novel merging property which we call *bounded difference δ-merging network*. This network mergers the outputs of two counting networks which have a difference of at most δ. The contention measurement is done using the recursive method introduced in [9]. The rest of the paper is organized as follows. In Section 2 we give the necessary definitions, in Section 3 we present the construction of a bounded difference δ-merging network, and in section 4 we present the construction of our counting network. Finally, in Section 5 we give our concluding remarks and present some open problems.

2. Definitions

We denote an integer sequence with a capital letter, e.g. X, and its elements with small letters e.g. x_i. The first index of a sequence is 0. Let $\Sigma(X)$ denote the sum of all the elements of X. From know on whenever we say that we compare two sequences X and Y we mean that we compare their sums, that is, we actually compare $\Sigma(X)$ and $\Sigma(Y)$. Take a sequence X of length (width) p. We say that X has the step property, or alternatively X is step, if $0 \leq x_i - x_j \leq 1$, for any i, j, $0 \leq i < j \leq p-1$. We say that X has the k-smooth property, or alternatively X is k-smooth, if $0 \leq |x_i - x_j| \leq k$, for any $0 \leq i, j \leq p-1$

Let b be a q-input, p-output balancer. Let X be the sequence (input sequence) of width q such that x_i is the number of tokens received on the ith wire of b, for all $0 \leq i \leq q-1$. In a similar way we define the sequence Y (output sequence) for the output wires. Let the *state* of b at a given time be defined as the collections of tokens received on its input and left from its output wires. A state of b is *quiescent* if $\Sigma(X) = \Sigma(Y)$; that is, the number of tokens that entered the balancer is equal to the number of tokens that left it. The following formal safety, liveness and step properties are required for b: (1) In any state, $\Sigma(X) \geq \Sigma(Y)$ (b never creates output tokens). (2) Given any finite number of input tokens to b it reaches within a finite amount of time a quiescent state (b never "swallows" input tokens). (3) In any quiescent state Y has the step property.

Let B be any balancing network of input width t and output width w. Let X and Y be the input and output se-

quences, respectively, of B. The state of B is defined as the collection of states of all its component balancers. Similarly to the balancer, the state of B is quiescent if $\Sigma(X) = \Sigma(Y)$. As in the balancer, we require B to have the safety and liveness properties. For the rest of the paper we will assume that the balancing networks we consider are in quiescent state.

A counting network is a balancing network such that its output sequence has the step property. We will denote a counting network of input width t and output width w as $C(t, w)$.

A bounded difference δ-merging network is a balancing network of equal input and output width, whose input sequence can be divided into two equal length subsequences A and B such that its output sequence has the step property whenever A and B both have the step property and $0 \leq \Sigma(A) - \Sigma(B) \leq \delta$. That is the difference between A and B is bounded by δ. We denote such a network of width w as $M(w, \delta)$. We will refer to the sequences A and B as the first and second input sequence, respectively, of $M(w, \delta)$.

On an MIMD shared memory multiprocessor machine, a balancing network B is implemented as a shared data structure, where balancers are records and wires are pointers from one record to another. Each of the machine's n asynchronous processors runs a program that repeatedly traverses the data structure from some input pointer to some output pointer, each time shepherding a new token through the network. Tokens generated by processor p_l, $l \in \{0, \ldots, n-1\}$, enter the network on input wire l mod t. The limitation on the number of concurrent processors implies a limitation on the number of tokens concurrently traversing the network at any given time: $\Sigma(X) - \Sigma(Y) \leq n$.

Consider an execution of B entering a quiescent state after m tokens pass through it. Each time a token passes through a balancer, all tokens pending at this balancer incur a *stall* step, modeling their delay due to contention with each other. The number of stall steps has been introduced in [7] as a measure of contention. The contention incurred by the traversal of m tokens through the network B at concurrency n, denoted $cont(m, n, B)$, is the maximum number of stalls, over all possible executions, induced by an adversary scheduler. The amortized contention of the network B at concurrency n, denoted $cont(n, B)$, is the limit supremum of $cont(m, n, B)$ divided by m, as m goes to infinity.

3. A Bounded Difference δ-Merging Network

In this section we present the construction of a bounded difference δ-merging network $M(w, \delta)$ where $w = p2^l$, $\delta = 2^k$, $p \geq 1$, $l \geq 2$, and $1 \leq k < l$. Let A and B denote the first and second input sequences, respectively, of $M(w, \delta)$ and let Y denote its output sequence.

The construction is by induction on δ. For the base case we have $\delta = 2$ and the network $M(w, 2)$ consists from $w/2$ 2-input, 2-output balancers $b_0, \ldots, b_{w/2-1}$. The first and second input wires of balancer b_i are connected to b_{i-1} and a_i, respectively, and its first and second output wires are connected to y_{2i-1} and y_{2i}, respectively, for $1 \leq i \leq w/2 - 1$. The first and second input wires of balancer b_0, are connected to a_0 and $b_{w/2-1}$, respectively, and its first and second output wires are connected to y_0 and y_{w-1}, respectively.

For the inductive case $\delta > 2$, the network $M(w, \delta)$ is constructed as follows (see figure 2). We take two copies of the network $M(w/2, \delta/2)$ denoted as $M_0(w/2, \delta/2)$ and $M_1(w/2, \delta/2)$. The first and second input sequences of $M_0(w/2, \delta/2)$ are connected to the even subsequences of A and B, respectively. The first and second input sequences of $M_1(w/2, \delta/2)$ are connected to the odd subsequences of A and B, respectively. Next, we take a copy of the network $M(w, 2)$. The first and second input sequences of $M(w, 2)$ are connected to the output sequences of $M_0(w/2, \delta/2)$ and $M_1(w/2, \delta/2)$, respectively. The output sequence of $M(w, 2)$ is connected to the sequence Y. This completes the construction. Next, we show the correctness of $M(w, \delta)$, then we calculate its depth and we estimate its contention.

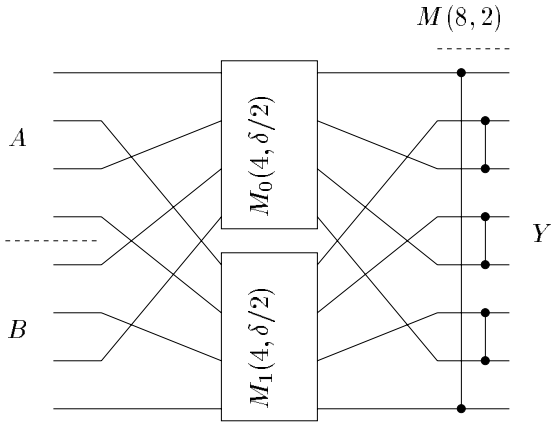

Figure 2. The network $M(8, \delta)$.

Proposition 3.1 *$M(w, \delta)$ is a bounded difference δ-merging network.*

Sketch of proof: First we examine the base case $\delta = 2$. If the difference between the input sequences A and B is exactly 0 or 1 then the balancers do not affect the input sequences and the output sequence Y has the step property. If the actual difference is exactly 2 then there are two subcases. In the first subcase the two input sequences form a 1-smooth sequence and one of the balancers $b_1, \ldots, b_{w/2-1}$

makes the output sequence to have the step property. In the second subcase the two input sequences form a 2-smooth sequence and the balancer b_0 makes the output sequence to have the step property.

Next, we examine the inductive case $\delta > 2$. Since the difference between the sequences A and B is at most δ we have that the difference between their even subsequences is at most $\delta/2$, and similarly the difference between their odd subsequences is at most $\delta/2$. Furthermore, these subsequences have the step property. Therefore, by the induction hypothesis, the outputs of networks $M_0(w/2, \delta/2)$ and $M_0(w/2, \delta/2)$ have the step property. Since in a sequence the even subsequence is greater by one or equal than the odd subsequence, we have that the output sequence of network $M_0(w/2, \delta/2)$ is bigger by at most two or equal to the output sequence of network $M_1(w/2, \delta/2)$. Therefore the output sequence of network $M(w, 2)$ has the step property, as needed. ■

Proposition 3.2 $depth(M(w, \delta)) = \lg \delta$.

Sketch of proof: We solve the recurrence $depth(M(w, \delta)) = depth(M(w/2, \delta/2)) + depth(M(w, 2))$. For the base case we have $depth(M(w, 2)) = 1$. ■

Proposition 3.3 $cont(m, n, M(w, \delta)) \leq m(2n/w - 1) \lg \delta$

Sketch of proof: Let m_0 and m_1 denote the number of tokens that enter the network $M_0(w/2, \delta/2)$ and $M_1(w/2, \delta/2)$, respectively. The construction guarantees concurrency $n/2$ for each of these networks. We solve the recurrence $cont(m, n, M(w, \delta)) \leq cont(m_0, n/2, M_0(w/2, \delta/2)) + cont(m_1, n/2, M_1(w/2, \delta/2)) + cont(m, n, M(w, 2))$. For the base case we have $cont(m, n, M(w, 2)) = m(2n/w - 1)$, since the contention of a balancer with concurrency n which traverse m tokens is equal to $m(n - 1)$. ■

4. A Counting Network

In this Section, we present a counting network $C(t, w)$ where $t = 2^k$, $w = p2^l$, $p, l \geq 1$, and $1 \leq k \leq l$. Let X and Y denote the input and output sequences, respectively, of $C(t, w)$.

The construction is by induction on t. For the base case we have $t = 2$ and the network $C(2, w)$ is just a 2-input, w-output balancer. For the inductive case $t > 2$ the network $C(t, w)$ is constructed as follows (see figure 3). We take $t/2$ 2-input, 2-output balancers $b_0, \ldots, b_{t/2-1}$. The first and second input wires of balancer b_i are connected to x_{2i} and x_{2i+1}, respectively, for all $0 \leq i \leq t/2 - 1$. Next, we take two copies of $C(t/2, w/2)$ denoted as $C_0(t/2, w/2)$ and $C_1(t/2, w/2)$. The first and second output wires of balancer b_i are connected to the ith input of the network $C_0(t/2, w/2)$ and $C_1(t/2, w/2)$, respectively, for all $0 \leq i \leq t/2 - 1$. Next, we take the bounded difference $t/2$-merging network $M(w, t/2)$ described in section 3. The first and second input sequences of network $M(w, t/2)$ are connected to the output sequences of the networks $C_0(t/2, w/2)$ and $C_1(t/2, w/2)$, respectively. The output sequence of $M(w, t/2)$ is connected to the sequence Y. This completes the construction. Next, we show the correctness of $C(t, w)$, then we calculate its depth and we estimate its contention.

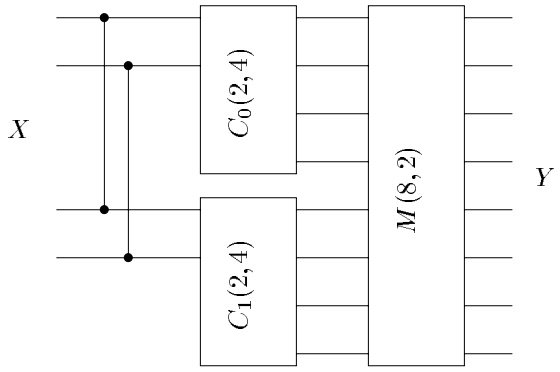

Figure 3. The network $C(4, 8)$.

Theorem 4.1 *$C(t, w)$ is a counting network.*

Sketch of proof: For the base case $t = 2$ the network $C(2, w)$ is obviously a counting network. For the inductive case $t > 2$ we have the following. The balancers $b_0, \ldots, b_{t/2-1}$ make the input sequence of network $C_0(t/2, w/2)$ to be greater by at most $t/2$ or equal to the input sequence of network $C_1(t/2, w/2)$. By the induction hypothesis, the output sequences of these two networks have the step property and furthermore their output sequences have the same difference as their input sequences. Therefore, by Proposition 3.1, we have that the output sequence of $M(w, t/2)$ has the step property, as needed. ■

Theorem 4.2 $depth(\mathcal{S}_{t,w}) = (\lg^2 t + \lg t)/2$

Sketch of proof: We solve the recurrence $depth(C(t, w)) = 1 + depth(C(t/2, w/2)) + depth(M(w, t/2))$ by using the result of Proposition 3.2. For the base case we have $depth(C(2, w)) = 1$. ■

Theorem 4.3 $cont(n, C(t, w)) \leq (n/w - 1/2) \lg^2 t + (2n/t - n/w - 1/2) \lg t$

Sketch of proof: Let m_0 and m_1 denote the number of tokens that enter the network $C_0(t/2, w/2)$ and

$C_1(t/2, w/2)$, respectively. Let L denote the first layer of balancers. We solve the recurrence $cont(m, n, C(t, w)) \leq cont(m, n, L) + cont(m_0, n/2, C_0(t/2, w/2)) + cont(m_1, n/2, C_1(t/2, w/2)) + cont(m, n, M(w, t/2))$. In order to do so we use Proposition 3.3. We also have $cont(m, n, L) \leq m(2n/t - 1)$. For the base case we have $cont(m, n, C(2, w)) \leq m(n - 1)$. Finally we take $cont(m, C(t, w)) = \lim_{m\to\infty}(cont(m, n, C(t, w))/m)$. ■

5. Concluding Remarks and Open Problems

We presented a counting network construction with t input wires and w output wires. where $t = 2^k$ and $w = p2^l$, and $k \leq l$. This is one of a very few constructions known whose output width is *not* a power of two [1, 6, 9].

Several interesting questions remain. Is it possible to extend our construction to arbitrary input and output widths, other than multiples of a power of two? It follows from impossibility results in [1, 5] that appropriate sets of balancer types would have to be used for such extension. Using such larger balancers is often expected to cause a reduction in depth (see [8, 9]). What would be a trade-off between depth and contention in this situation? Can the combinatorial techniques in [5] be used to show impossibility results on constructible widths for bounded difference δ-merging networks?

We believe that our counting network will allow a significant improvement in performance when used in real shared memory multiprocessors, over previously used counting network constructions [1, 3, 6, 8, 9]. To verify our belief, we are currently implementing a software simulation of our counting network construction in a general asynchronous multiprocessor. We hope this simulation will enable us to evaluate the performance of our construction as measured by contention. In our simulations, we fix a particular input width and compare the counting network introduced in this paper to the bitonic and periodic counting networks [3] of the same width. Preliminary experimental investigations reveal that for appropriately chosen values of the parameter w, especially when it is taken to be of order $\Theta(t \lg t)$, the counting network resulting from our construction significantly outperforms the other two under identical concurrency conditions.

Acknowledgements:

We thank Sergio Rajsbaum for helpful comments on an earlier draft of this paper, Maurice Herlihy for useful discussions, and Pearl Tsai for useful remarks.

References

[1] E. Aharonson and H. Attiya, "Counting Networks with Arbitrary Fan-Out," *Distributed Computing*, Vol. 8, pp. 163–169, 1995.

[2] W. Aiello, R. Venkatesan and M. Yung, "Coins, Weights and Contention in Balancing Networks," *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing*, pp. 193–205, August 1994.

[3] J. Aspnes, M. Herlihy and N. Shavit, "Counting Networks," *Journal of the ACM*, Vol. 41, No. 5, pp. 1020–1048, September 1994.

[4] K. E. Batcher, "Sorting Networks and their Applications," *Proceedings of AFIPS Spring Joint Computer Conference*, pp. 307–314, 1968.

[5] C. Busch and M. Mavronicolas, "A Combinatorial Treatment of Balancing Networks," *Journal of the ACM*, Vol. 43, No. 5, pp. 749–839, September 1996.

[6] C. Busch, N. Hardavellas and M. Mavronicolas, "Contention in Counting Networks," *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing*, pp. 404, August 1994.

[7] C. Dwork, M. Herlihy and O. Waarts, "Contention in Shared Memory Algorithms," *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pp. 174–183, May 1993.

[8] E. W. Felten, A. LaMarca and R. Ladner, "Building Counting Networks from Larger Balancers," Technical Report 93-04-09, Department of Computer Science and Engineering, University of Washington, April 1993.

[9] N. Hardavellas, D. Karakos and M. Mavronicolas, "Notes on Sorting and Counting Networks," *Proceedings of the 7th International Workshop on Distributed Algorithms (WDAG-93)*, Lecture Notes in Computer Science, Vol. 725 (A. Schiper, ed.), Springer-Verlag, pp. 234–248, Lausanne, Switzerland, September 1993.

[10] D. Knuth, *Sorting and Searching*, Volume 3 of *The Art of Computer Programming*, Addison-Wesley, 1973.

[11] N. Shavit and A. Zemach, "Diffracting Trees," *Proceedings of the 6th Annual ACM Symposium on Parallel Algorithms and Architectures*, pp. 167–176, June 1994.