Efficient Quantum Computation with Probabilistic Quantum Gates

L.-M. Duan[1] and R. Raussendorf[2]

[1]FOCUS Center and MCTP, Department of Physics, University of Michigan, Ann Arbor, Michigan 48109-1120, USA
[2]Institute for Quantum Information, California Institute of Technology, Pasadena, California 91125, USA

(Received 28 February 2005; published 17 August 2005)

With a combination of the quantum repeater and the cluster state approaches, we show that efficient quantum computation can be constructed even if all the entangling quantum gates only succeed with an arbitrarily small probability p. The required computational overhead scales efficiently both with $1/p$ and n, where n is the number of qubits in the computation. This approach provides an efficient way to combat noise in a class of quantum computation implementation schemes, where the dominant noise leads to probabilistic signaled errors with an error probability $1 - p$ far beyond any threshold requirement.

DOI: 10.1103/PhysRevLett.95.080503 PACS numbers: 03.67.Lx, 03.67.Pp, 03.67.Mn

The celebrated threshold theorem in quantum computation has assured that, if the amount of noise per quantum gate is less than a small value, reliable quantum computation can be efficiently performed [1]. In terms of implementation, however, the experimental noise is typically orders of magnitude large than the required threshold value. To overcome this problem, a practical route to noise reduction is by exploitation of certain properties of the noise. In carefully designed implementation schemes, the dominant noise leads to specific types of errors only, which can be corrected much more efficiently. Here, we consider an important noise model, which is relevant for several experimental approaches to quantum computation with the use of photons or trapped atoms or ions as the qubits [2–11]. In these schemes, the dominant noise leads to significant failure probability for the entangling gates only, and a gate failure is always signaled through built-in fast photon detections during the gate operation. The success probability p for each entangling gate is rather small for some typical experimental systems [4,10,11]. It is hard to use the standard methods of error correction in the considered scenario, because the error probability $1 - p$ (close to the unity) is simply too large.

Naively, if a gate succeeds only with a certain probability p, one cannot have efficient computation, as the overall success probability (efficiency) scales down exponentially as p^n with the number n of gates. However, in this Letter we show that efficient quantum computation can be constructed with the required computational overhead (such as the computation time or the repetition number of the entangling gates) scaling up slowly (polynomially) with both n and $1/p$. The demonstration of this result combines the ideas from the quantum repeater schemes [12,13] and the cluster state approach to computation [14,15]. It has been shown in quantum repeater schemes that, with probabilistic entangling operations, one can construct scalable quantum communication and Greenberger-Horne-Zeilinger correlations [13,16]. Recently, it has also been demonstrated in the context of linear optics computation [2] that the threshold requirement on the probability p for the entangling gates can be significantly improved using the cluster state approach to quantum computation [3,5,7,8]. In particular, Ref. [8] shows that for construction of one-dimensional (1D) cluster states, to get efficient scaling with n, in principle no threshold value is needed on p, although in practice p is still required to be sufficiently large as the computational overhead in that scheme has inefficient (superexponential) scaling with $1/p$. Compared with these previous results, here we have the following two advances: (i) We propose a probabilistic computation scheme which has efficient scaling with both n and $1/p$. This improvement is substantial as in current experiments $1/p$ is large [10,11]. (ii) Through explicit construction, we also demonstrate efficient scaling of the computational overhead for generation of the two-dimensional (2D) cluster states which are critical for realization of universal quantum computation.

To be more specific, we assume in this Letter that one can reliably perform two-qubit controlled phase flip (CPF) gates with a small success probability p, although the basic ideas here also apply for other kinds of entangling gates. We neglect the noise for all the single-bit operations, which is well justified for typical atomic or optical experiments. Our basic steps are as follows: First, we show how to efficiently prepare 1D cluster state from probabilistic CPF gates. Then, we give a construction to efficiently generate 2D cluster states from 1D chains. Efficient preparation of 2D cluster states, together with simple single-bit operations, realizes universal quantum computation.

With respect to a given lattice geometry, the cluster state is defined as coeigenstates of all the operators $A_i = X_i \prod_j Z_j$, where i denotes an arbitrary lattice site and j runs over all the nearest neighbors of the site i. The X_i and Z_j denote, respectively, the Pauli spin and phase flip operators on the qubits at the sites i, j. In our construction of lattice cluster states with probabilistic CPF gates, we will make use of the following three properties of the cluster states: (i) If we have two chains of cluster states each with n qubits, we can join them to form a 1D cluster

state of $2n$ qubits by successfully applying a CPF gate on the end qubits of the two chains. (ii) If we destroy the state of an end qubit of an n-qubit cluster chain, for instance, through an unsuccessful attempt of the CPF gate, we can remove this bad qubit by performing a Z measurement on its neighboring qubit, and recover a cluster state of $n-2$ qubits. (iii) We can shrink a cluster state by performing X measurements on all the connecting qubits [see Fig. 1(c)]. These three properties of the cluster states, illustrated in Fig. 1, can be conveniently explained from their above definition [15,17].

If we have generated two sufficiently long cluster chains each of n_0 qubits, we can just try to connect them through a probabilistic CPF gate. If this attempt fails, through the property (ii), we can recover two (n_0-2)-qubit cluster chains through a Z measurement, and try to connect them again. As one continues with this process, the average number of the qubits in the connected chain is then given by $n_1 = \sum_{i=0}^{n_0/2} 2(n_0-2i)p(1-p)^i \simeq 2n_0 - 4(1-p)/p$, where the last approximation is valid when $e^{-n_0 p/2} \ll 1$. So the average chain length goes up if $n_0 > n_c \equiv 4(1-p)/p$. We can iterate these connections to see how the computation overhead scales with the qubit number n. We measure the computation overhead in terms of the total computation time and the total number of attempts for the CPF gates. For the rth ($r \geq 1$) round of successful connection, the chain length n_r, the total preparation time T_r, and the total number of attempts M_r scale up, respectively, by the recursion relations $n_r = 2n_{r-1} - n_c$, $T_r = T_{r-1} + t_a/p$, and $M_r = 2M_{r-1} + 1/p$. In writing the recursion relation for T_r, we have assumed that two cluster chains for each connection are prepared in parallel, and we neglect the time for single-bit operations (t_a denotes the time for each attempt of the CPT gate). From the above recursion relations, we conclude that if we can prepare cluster chains of n_0 ($n_0 > n_c$) qubits in time T_0 with M_0 attempts of the probabilistic gates, for a large cluster state, the preparation time T and the number of attempts M scale with the chain length n as $T(n) = T_0 + (t_a/p)\log_2[(n-n_c)/(n_0-n_c)]$ and $M(n) = (M_0 + 1/p)(n-n_c)/(n_0-n_c) - 1/p$.

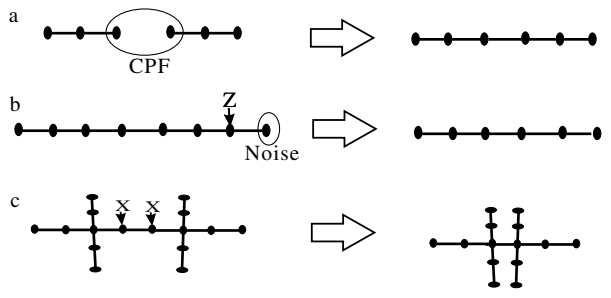

FIG. 1. Illustration of the three properties of the cluster states which are important for our construction of such states with the probabilistic entangling gates: (a) extend cluster states with CPF gates; (b) recover cluster states by removing bad qubits; (c) shrink cluster states for more complicated links.

In the above, we have shown that if one can prepare cluster chains longer than some critical length n_c, one can generate large scale 1D cluster states very efficiently. The problem then reduces to how to efficiently prepare cluster chains up to the critical length n_c. If one wants to prepare an n-qubit cluster chain, we propose to use a repeater protocol which divides the task into $m = \log_2 n$ steps: For the ith ($i = 1, 2, \ldots, m$) step, we attempt to build a 2^i-bit cluster state by connecting two 2^{i-1}-bit cluster chains through a probabilistic CPF gate. If such an attempt fails, we discard all the qubits and restart from the beginning [18]. For the ith step, the recursion relations for the preparation time T_i and the number of attempts M_i are given by $T_i = (1/p)(T_{i-1} + t_a)$ [19] and $M_i = (1/p)(2M_{i-1} + 1)$, which, together with $T_1 = t_a/p$ and $M_1 = 1/p$, give the scaling rules $T(n) \simeq t_a(1/p)^{\log_2 n}$ and $M(n) \simeq (2/p)^{\log_2 n}/2$. The cost is more significant, but it is still a polynomial function of n. To construct a n-qubit cluster chain, in total we need $n-1$ successful CPF gates. In a direct protocol, we need all these attempts to succeed simultaneously, which gives the scaling $T(n) \propto M(n) \propto (1/p)^{n-1}$. By dividing the task into a series of independent pieces, we improve the scaling with n from exponential to polynomial (for $n \leq n_c$).

To generate a cluster chain of a length $n > n_c$, we simply combine the above two protocols. First, we use the repeater protocol to generate n_0-qubit chains with $n_0 > n_c$. Then it is straightforward to use the connect-and-repair protocol to further increase its length. For instance, with $n_0 = n_c + 1$ (which is a reasonable close-to-optimal choice), the overall scaling rules for T and M are (for $n > n_c$)

$$T(n) \simeq t_a(1/p)^{\log_2(n_c+1)} + (t_a/p)\log_2(n-n_c), \quad (1)$$

$$M(n) \simeq (2/p)^{\log_2(n_c+1)}(n-n_c)/2. \quad (2)$$

As the critical length is $n_c \simeq 4/p$, T and M in our protocol scale with $1/p$ as $(1/p)^{\log_2(4/p)}$, which is much more efficient than the superexponential scaling $(1/p)^{4/p}$ in the previous work.

We have shown that for any success probability p of the probabilistic entangling gate, 1D cluster states of arbitrary length can be created efficiently. For universal quantum computation, however, such 1D cluster states are not sufficient. They need to be first connected and transformed into 2D cluster states (for instance, with a square lattice geometry) [15]. It is not obvious that such a connection can be done *efficiently*. First, in the connect-and-repair protocol, when an attempt fails, we need to remove the end qubits and all of their neighbors. This means that in a 2D geometry the lattice shrinks much faster to an irregular shape in the events of failure. Furthermore, a more important obstacle is that we need to connect much more boundary qubits if we want to join two 2D cluster states. For

instance, for a square lattice of n qubits, the number of boundary qubits scales as $\sqrt{n}$ (which is distinct from a 1D chain). If we need to connect all the corresponding boundary qubits of the two parts, the overall success probability is exponentially small.

To overcome this problem, we introduce a method which enables efficient connection by attaching a long leg (a 1D cluster chain) to each boundary qubit of the 2D lattice. The protocol is divided into the following steps: First, we try to build a ''+-shape'' cluster state by probabilistically connecting two cluster chains each of length $2n_l + 1$ (the value of n_l will be specified below). This can be done through the probabilistic CPF gate together with a simple Hardmard gate H and an X measurement, as shown in Fig. 2(a) and explained in its caption. With on average $1/p$ repetitions, we get a +-shape state with the length of each of the four legs given by n_l. We use the +-shape state as the basic building blocks of large scale 2D cluster states. In the +-shape state, we have attached four long legs to the center qubit. The leg qubits serve as ancilla to generate near-deterministic connection from the probabilistic CPF gates. The critical idea here is that if we want to connect two center qubits, we always start the connection along the end qubits of one of the legs (see illustration in Fig. 2). If such an attempt fails, we can delete two end qubits and try the connection again along the same legs. If the leg is sufficiently long, we can almost certainly succeed before we reach (destroy) the center qubits. When we succeed, and if there are still redundant leg qubits between the two center ones, we can delete the intermediate leg qubits by performing simple single-bit X measurements on all of them [see Figs. 1(c) and 2 for the third property of the cluster state]. With such a procedure, we can continuously connect the center qubits and form any complex lattice geometry [see the illustration for construction of the square lattice state in Figs. 2(b) and 2(c)]. What is important here is that after each time of connection of the center qubits, in the formed new shape, we still have the same length of ancillary legs on all the boundary qubits, which enables the succeeding near-deterministic connection of these new shapes.

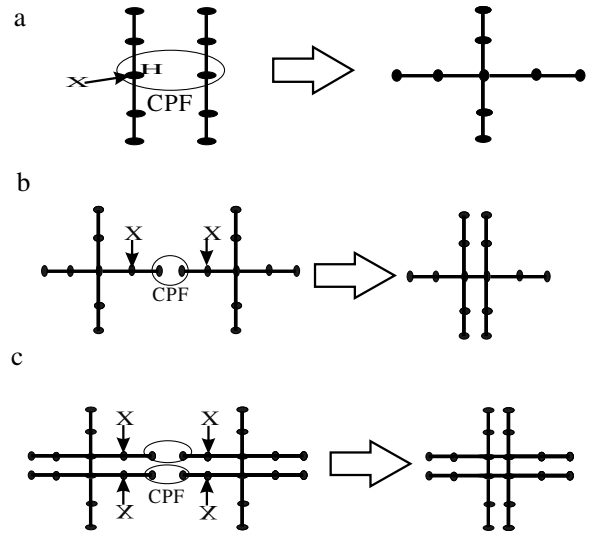

FIG. 2. Illustration of the steps for construction of the 2D square lattice cluster states from a set of cluster chains. (a) Construction of the basic +-shape states from cluster chains by applying first a Hardmard gate H on the middle qubit of one chain, and then a CPF gate to connect the two middle qubits, and finally a X measurement on one middle qubit to remove it. (b),(c) Construction of the square lattice cluster state from the +-shape states through probabilistic CPF gates along the legs and X measurements to remove the remaining redundant qubits.

Now we investigate for the 2D case how the computational overhead scales with the size of the cluster state. If the ancillary legs have length n_l, for each connection of two center qubits, we can try at most $n_l/2$ times of the probabilistic CPF gates, and the overall success probability is given by $p_c = 1 - (1 - p)^{n_l/2}$. If we want to build a square lattice cluster state of N qubits, we need about $2N$ times of connections of the center qubits (there are about $2N$ edges in an N-vertex square lattice). The probability for all these connections to be successful is given by p_c^{2N}. We require that this overall success probability is sufficiently large with $p_c^{2N} \geq 1 - \epsilon$, where ϵ is a small number characterizing the overall failure probability. From that requirement, we figure out that $n_l \simeq (2/p)\ln(2N/\epsilon)$. To construct a square lattice cluster state of N qubits, we need to consume N +-shape states, and each of the latter requires on average $2/p$ cluster chains with a length of $2n_l + 1$ qubits. So we need in total $(2N/p)(2n_l + 1)$-bit cluster chains, which can be prepared in parallel with $(2N/p)M(2n_l + 1)$ CPF attempts within a time period $T(2n_l + 1)$ [see Eqs. (1) and (2) for expressions of the $M(n)$ and $T(n)$]. This gives the resources for preparation of all the basic building blocks (the chains). Then we need to connect these blocks to form the square lattice. We assume that the connection of all the building blocks are done in parallel. The whole connection takes on average $2N/p$ CPF attempts, and consumes a time at most $t_a/p\ln(2N/\epsilon)$. Summarizing these results, the temporal and the operational resources for preparation of an N-bit square lattice cluster state are approximately given by

$$T(N) \simeq t_a(1/p)^{\log_2(4/p-3)} + \frac{t_a}{p}\log_2\left(\frac{4}{p}[\ln(2N/\epsilon) - 1]\right) + \frac{t_a}{p}\ln(2N/\epsilon), \quad (3)$$

$$M(N) \simeq (2/p)^{2+\log_2(4/p-3)}N[\ln(2N/\epsilon) - 1] + 2N/p. \quad (4)$$

In the 2D case, the temporal and the operational overheads still have very efficient scaling with the qubit number N, logarithmically for $T(N)$ and $N\ln(N)$ for $M(N)$. Their scalings with $1/p$ are almost the same as in the 1D case except an additional factor of $1/p^2$ for $M(N)$. Through

some straightforward variations of the above method, it is also possible to efficiently prepare any complicated graph state using probabilistic CPF gates [17]. This shows that in principle we do not need to impose any threshold on the success probability of the CPF gates for efficient quantum computation.

Before ending the Letter, we would like to add a few remarks on other sources of noise that have not been taken into account in the above discussions. If each CPF gate has some small additional infidelity error, one might wonder whether such an error scales up with the large number of attempts $M(N)$. That is actually not the case. Most of the CPF attempts have failed, and all the failed CPF gates have no contribution to the final state infidelity. In practice, we may be more concerned about the temporal overhead $T(N)$ than the operational overhead $M(N)$. Each qubit has a finite coherence time and we need to finish all the CPF attempts within such a time scale. For typical probabilistic entangling experiments with atoms [10,11], the time t_a for each CPF attempt is about 100 ns, while the qubit coherence time is usually longer than 1 s. If we take the success probability $p \sim 0.1$, Eq. (3) gives $T(N) \sim 1.6 \times 10^5 t_a \sim$ 16 ms for any large N [20], which is still well within the qubit coherence time.

In summary, we have shown that cluster states in two dimensions can be generated using probabilistic CPF gates with efficient scaling in both the qubit number and the inverse of the success probability. This result opens up a prospect to realize efficient quantum computation with probabilistic entangling gate operations. Such a prospect is relevant for several experimental systems involving atoms, ions, or photons [2,4,10,11], with ongoing efforts towards probabilistic quantum information processing.

L. M. D. thanks Chris Monroe for helpful discussions. This work was supported by NSF grants (0431476 and EIA-0086038), the ARDA under ARO contracts, the A. P. Sloan Foundation, and the visitor program of MCTP.

Note added.—Recently, we became aware that in the latest version of Ref. [8] the authors have also made some interesting improvement of their construction efficiency of the 1D cluster state to overcome the inefficient scaling.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2000).

[2] E. Knill, R. Laflamme, and G. Milburn, Nature (London) **409**, 46 (2001).

[3] N. Yoran and B. Reznik, Phys. Rev. Lett. **91**, 037903 (2003).

[4] L.-M. Duan and H. J. Kimble, Phys. Rev. Lett. **92**, 127902 (2004); **90**, 253601 (2003); L.-M. Duan, B. Wang, and H. J. Kimble, quant-ph/0505054.

[5] M. A. Nielsen, Phys. Rev. Lett. **93**, 040503 (2004).

[6] L.-M. Duan, B. B. Blinov, D. L. Moehring, and C. Monroe, Quantum Inf. Comput. **4**, 165 (2004).

[7] D. E. Browne and T. Rudolph, quant-ph/0405157.

[8] S. D. Barrett and P. Kok, Phys. Rev. A **71**, 060310(R) (2005).

[9] X. Zou and W. Mathis, quant-ph/0401042; Y. L. Lim, A. Beige, and L. C. Kwek, quant-ph/0408043.

[10] B. B. Blinov, D. L. Moehring, L.-M. Duan, and C. Monroe, Nature (London) **428**, 153 (2004).

[11] A. Kuzmich *et al.*, Nature (London) **423**, 731 (2003); C. H. van der Wal *et al.*, Science **301**, 196 (2003); W. Jiang *et al.*, Phys. Rev. A **69**, 043819 (2004); D. N. Matsukevich and A. Kuzmich, Science **306**, 663 (2004).

[12] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, Phys. Rev. Lett. **81**, 5932 (1998).

[13] L. M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, Nature (London) **414**, 413 (2001).

[14] H. J. Briegel and R. Raussendorf, Phys. Rev. Lett. **86**, 910 (2001).

[15] R. Raussendorf and H. J. Briegel, Phys. Rev. Lett. **86**, 5188 (2001).

[16] L.-M. Duan, Phys. Rev. Lett. **88**, 170402 (2002).

[17] R. Raussendorf, D. E. Browne, and H. J. Briegel, Phys. Rev. A **68**, 022312 (2003).

[18] If we do not immediately discard all the qubits, but instead do some repairing and reconnection as described in the previous paragraph, the protocol will be somewhat more efficient. But for $n \leq n_r$, such repairing has no significant influence on the basic scaling rules.

[19] For this recursion relation on T_i, we have assumed that we use only one copy of the system with n qubits (no ancillary qubits) and that, if we fail, we start over from the first step. However, if we allow very many copies of such n-qubit systems on which we operate in parallel, in principle we can get much more efficient scaling for T_i. We try the chain connection in parallel in all these n-qubit systems, discard the systems with failure events, and continue the next-step connection for the systems where all the previous connections are successful. In this way, one connection (one length doubling of the qubit chain) takes only one time step t_a, so we have $T_i = T_{i-1} + t_a$. This recursion relation will lead to a significant improvement in the total required preparation time: the scaling of $T(N)$ [or $T(n)$] with $1/p$ in Eq. (3) [or Eq. (1)] changes from $(1/p)^{\log_2(4/p)}$ to $(1/p)\log_2(4/p)$. Of course, the cost for this improvement is probably too expensive for practical experiments: to prepare n-qubit cluster states with the above parallel operations, we need to have roughly $n(1/p)^{\log_2(4/p)}$ ancillary qubits.

[20] For the quoted number, $T(N)$ is basically independent of N for $N/\epsilon < 10^{7000}$ due to the slow logarithmic scaling $\ln(2N/\epsilon)$.