Quantum Algorithms for Leader Election Problem in Distributed Systems

Pradeep Sarvepalli

pradeep@cs.tamu.edu

Department of Computer Science,

Texas A&M University

Outline

- Introduction to distributed systems
  - Model of distributed systems
  - Leader election in distributed systems
- Quantum distributed systems
  - Quantum resources
  - Model of quantum distributed systems
- Quantum leader election (QLE) algorithms
  - 2-party leader election
  - n-party leader election
  - QLE - Necessary and sufficient quantum resources

Outline- cont'd

- QLE due to Tani et al.
  - 2-party leader election
  - n-party leader election
- Open Issues & Conclusions

Introduction

- Distributed Systems
  - Processors connected by a communication network
  - Processors are loosely coupled more or less independent
  - In our case we assume no shared memory, clock
- Anonymous networks
  - Processors do not have unique identifiers
- Synchronous networks
  - Processors send and receive messages
  - Followed by a local computation
  - Bounds on timing delays known

Leader Election in Distributed Systems

- A leader in a distributed system
  - coordinates the activities
  - reduces complexity of tasks
  - helps in fault tolerance
- Leader Election in a distributed system of n processors
  - Each processor has a local variable $Elected$ initialized to 0
  - Each processor runs the exact same algorithm A
  - On termination exactly one processor should have the variable $Elected$ set to 1

Leader Election in Anonymous Networks

- Anonymous networks
  - Processors do not have unique identifiers
- In anonymous networks there is no deterministic algorithm for electing a leader
- The main reason is that the processors are indistinguishable and this symmetry prevents leader election
- One solution to break the symmetry is to assume that the processors are provided with a fair coin

A Randomized Leader Election Algorithm

- 2-party
  - Each party flips a coin and communicates the outcome to the other party
  - The party which obtained heads is elected leader
  - If only one processor gets a head then there is no problem
  - If both get heads or tails then they repeat until there is only one head
- In practice quite efficient, expected running time is 2 rounds
- However, this algorithm will not always terminate

Quantum Distributed Systems

- The primary difference between quantum and classical distributed systems is the use of entangled qubits and/or quantum channels
- Quantum networks have at least three models depending on how they communicate and the presence or absence of entangled data
  - Processors communicate qubits
  - Processors do not share entangled pairs, communicate bits
  - Processors share entangled pairs, communicate qubits

Quantum Resources - Entangled States

- Maximally entangled states

$$GHZ_3 = |000\rangle + |111\rangle$$

- If we measured one qubit say the first one, we would get $|000\rangle$ or $|111\rangle$
- The resulting states are not entangled at all!!
- The entanglement is destroyed by one measurement
- In general the GHZ_n state is

$$GHZ_n = |0^{\otimes^n}\rangle + |1^{\otimes^n}\rangle$$

Quantum Resources - Entangled States

- Alternatively consider

$$W_3 = |100\rangle + |010\rangle + |001\rangle$$

- If we measure this state then with probability $2/3$ we would get $|010\rangle + |001\rangle = |0\rangle(|10\rangle + |01\rangle)$ and with probability $1/3$ get $|100\rangle$
- $|010\rangle + |001\rangle$ is still entangled
- W_3 state needs two measurements before we get a separable state
- In general the W_n state is

$$W_n = |100\ldots0\rangle + |01\ldots0\rangle + \cdots + |0\ldots01\rangle$$

Quantum Distributed Systems

- Processors connected by a communication network (classical/quantum)
- No shared memory
- No common clock
- Entangled qubits available (sometimes)
- Anonymity implies that the intial quantum state is invariant under permutation of processors

2-party Leader Election

- Let A, B share the state $|0_A 1_B\rangle + |1_A 0_B\rangle = |01\rangle + |10\rangle$
- Algorithm
  - Perform measurement on ith qubit
  - If 1, then elect itself as leader
- Illustration
  - The resulting state is $|01\rangle$ or $|10\rangle$
  - The complementary measurements of A, B ensure that there is no conflict and a leader is elected after the first round

n-party Leader Election

- Let the processors share the state

$$W_n = |10\ldots0\rangle + |010\ldots0\rangle + \cdots + |0\ldots01\rangle$$

$$W_n = |2^{n-1}\rangle + |2^{n-2}\rangle + \cdots + |2\rangle + |1\rangle$$

- Algorithm
  - Let each processors measure its qubit
  - If measurement is 1, then elect itself as leader

Quantum Leader Election Algorithm - D'Hondt et. al

Data: Entangled state W_n
Result: If elected leader then elected is set to 1
elected:=0;
m:=Measure ith qubit;
if *m=1* **then**
| elected=1;
end

Algorithm 1: QLE Algorithm

QLE- Some Questions

- Is the algorithm fair?
  - Does every processor get elected with the same probability?
- Are there any other entangled states that we can use for QLE?
- Are these quantum networks truly anonymous?
  - Does the use of W_n remove anonymity somehow?
- Can we elect a leader without entanglement?
- How does one share the entangled state W_n?

QLE- Some Questions

- Is the algorithm fair? Yes. Any processor is elected with probability $1/n$
- Are there any other entangled states that we can use for QLE? No
- Are these quantum networks truly anonymous? Yes. The initial shared quantum state is invariant under permutation
- Can we elect a leader without entanglement? No
- How does one share the entangled state W_n?

QLE - Tani et. al

- There was an alternate approach proposed by Tani et. al, which is more complete in the sense it addresses how to share the entanglement and other details
- Basic idea is same
  - Use entangled states which on measurement create asymmetry among the processors
- We will illustrate the algorithm with 2-party as it is easier to understand the key ideas

2-party QLE due to Tani et al.

- Each party prepares the state $R = (|0\rangle + |1\rangle)/\sqrt{2}$
- System state is $R_x R_y = |\psi\rangle = |00\rangle + |01\rangle + |10\rangle + |11\rangle$
- In a separate register each processor computes if both the bits are same
- Now the global state is

$$R_x R_y S_x S_y = (|00\rangle + |11\rangle)|11\rangle + (|01\rangle + |10\rangle)|00\rangle$$

- Note that the registers S_x and S_y are entangled

2-party QLE - cont'd

- Each processor measures its S register
- The state will collapse to either $(|00\rangle + |11\rangle)|11\rangle$ or $(|01\rangle + |10\rangle)|00\rangle$
- It does not matter who measures first
- If we get $(|01\rangle + |10\rangle)|00\rangle$, then we are done.
  - Let each processor measure its register R
  - We will get either $|01\rangle$ or $|10\rangle$ and an unique leader
- If we get $(|00\rangle + |11\rangle)|11\rangle$, then somehow we have to transform it to W_2 state i.e., $(|01\rangle + |10\rangle)$

2-party QLE - cont'd

- Each processor applies the unitary operation

$$U_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$$

- Now the state $|00\rangle + |11\rangle$ gets transformed to

$$(|0\rangle - i|1\rangle) \otimes (|0\rangle - i|1\rangle) + (-i|0\rangle + |1\rangle) \otimes (-i|0\rangle + |1\rangle)$$

$$|00\rangle - i|01\rangle - i|10\rangle + i^2|11\rangle + i^2|00\rangle - i|01\rangle - i|10\rangle + |11\rangle$$

$$= -i|01\rangle - i|10\rangle$$

2-party QLE - cont'd

- With the W_2 state in hand we can proceed to elect a leader as before
  - Let each processor measure its register R
  - We will get either $|01\rangle$ or $|10\rangle$ and an unique leader

n-party QLE

- The generalization is essentially the same idea but complicated
- A string $x = x_1 x_2 \ldots x_n$ of length bn is consistent if all substrings x_i are same
- Let each processor create the state $R_i = |0\rangle + |1\rangle$
- This gives the global state

$$R_1 \cdots R_n = \sum_{i=0}^{2^n - 1} |i\rangle$$

- Let each processor locally store in S_i the consistency of the global state

n-party QLE - cont'd

- We can partition the global state as

$$R_1 \cdots R_n S_1 \cdots S_n = (|0^{\otimes^n}\rangle + |1^{\otimes^n}\rangle)|1^{\otimes^n}\rangle + \sum_{i=1}^{2^n-2} |i\rangle|0^{\otimes^n}\rangle$$

- Again note that S_i are entangled
- Now let each processor measure its S register. We will get either

$$(|0^{\otimes^n}\rangle + |1^{\otimes^n}\rangle)|1^{\otimes^n}\rangle \text{ or } \sum_{i=1}^{2^n-2} |i\rangle|0^{\otimes^n}\rangle$$

n-party QLE - cont'd

- If we get $\sum_{i=1}^{2^n-2} |i\rangle|0^{\otimes^n}\rangle$, then each processor can measure its qubit R_i
- Because the states are inconsistent atleast one processor will measure 0 and the rest 1 or 0
- Promote those which have measured 1 to the next phase for leader election and discard the ones which have measured 0
- Thus we have reduced it to smaller leader election problem
- Worst case we will need $n-1$ phases

n-party QLE - cont'd

- If we get the GHZ_n

$$(|0^{\otimes^n}\rangle + |1^{\otimes^n}\rangle)|1^{\otimes^n}\rangle$$

  we have to transform it to an inconsistent state so that there is asymmetry in the global state

- If the number of parties k, initially n
  - even, then we apply the operator

$$U_k = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{-i\pi/k} \\ -e^{i\pi/k} & 1 \end{pmatrix}$$

n-party QLE - cont'd

- odd
  - We need an additional register T_i initialized to $|0\rangle$
  - Consider the global state $R_1 \ldots R_k T_1 \ldots T_k$
  - $T_i \mapsto R_i \oplus T_i$ and then apply V_k to $R_i T_i$

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \sqrt{R_k} & e^{i\pi/k}/\sqrt{2} \\ \frac{1}{\sqrt{2}} & 0 & -\sqrt{R_k}e^{-i\pi/k} & e^{-i\pi/k}/\sqrt{2} \\ \sqrt{R_k} & 0 & \frac{e^{-i\pi/2k} I_k}{i\sqrt{2}R_{2k}} & -\sqrt{R_k} \\ 0 & \sqrt{R_k+1} & 0 & 0 \end{pmatrix}$$

n-party QLE - cont'd

- The previous step always leads to an inconsistent state
- Once again each processor measures its qubits R_iT_i
- This time we select only those processors which have the maximum value in R_iT_i
- Because the states are inconsistent we are guaranteed that atleast some processor is discarded from the election
- Repeat this algorithm with the newer set

QLE 2 - (Sketch)

Result: If elected leader then $Elected$ is set to 1
$Elected := 0, Eligible := 1, S := 0;$
for $k \leftarrow n$ **to** 2 **do**
    if *Eligible=1* **then**
        Prepare $R = |0\rangle + |1\rangle$;
        Compute consistency of global state in S
        Measure S;
        if *S=1* **then**
            Transform into an inconsistent state;
        end
        Measure R;
        Discard if $R = 0$, Eligible:=0;
    end
end

Complexity of QLE 2

- Running time $O(n^3)$
- Communication complexity $O(n^4)$
- Quantum communication complexity $O(n^4)$
- Quantum round $\theta(n^2)$
- A modified algorithm exists with increased running time

Open Issues & Conclusions

- Quantum computing seems to be beneficial for some distributed tasks
- Can we show some equivalence between the two algorithms?
- How does one share the entangled state W_n for the D’Hondt algorithm?
  - What is the complexity of this algorithm taking into account the implementation details?
- Can the algorithm due to Tani et al. be simplified?
- Are there some good quantum algorithms for
  - Mutual exclusion
  - Fault tolerant consensus (Crash and Byzantine)

References

- References
  - “Leader Election and Distributed Consensus with Quantum Resources” by E. D’Hondt and P. Panangaden
  - “Exact Quantum Algorithms for the Leader Election Problem” by S. Tani, H. Kobyashi and K. Matsumoto

Questions ?

Questions ?

Thank You !