Quantum Distributed Consensus

Louie Helm

University of Texas at Austin

Supervisor: Christine Julien

Abstract: Distributed consensus is possible for asynchronous communication networks when assisted by quantum mechanical effects. Previous results are extended to show that consensus is possible even in the presence of byzantine process failures. This result directly contradicts the FLP impossibility result which states that distributed consensus is impossible when even one faulty networked process exists in the group.

PODC'08, August 18–21, 2008, p445, Toronto, Ontario, Canada. ACM 978-1-59593-989-
0/08/08.

1. DISTRIBUTED COMPUTING IS PHYSICAL

Distributed computing is concerned with what can be accomplished using systems of networked processors. As with all information processing devices, these systems are necessarily physical and ultimately bound by the laws of physics[Land88]. Therefore, when proving impossibility results and bounds on algorithmic efficiency, it only makes sense to consider the possibilities offered, not only by classical information processing, but also the effects of quantum mechanics[NC00]. Recent results have shown that shared quantum resources make problems like anonymous leader election and distributed consensus possible under conditions where no classical algorithm could perform the same tasks[DP06]. These results are extended here to show that shared quantum resources allow fault-tolerant consensus in direct contradiction of previous impossibility results[FLP85].

Although this research is largely self-contained, a basic understanding of distributed computing is assumed of the reader along with a passing understanding of linear algebra which is necessary to understand the reasoning behind the algorithm detailed in section 4. A thorough understanding of quantum mechanics is recommended, although a reference of the specific quantum mechanical notation required to follow later reasoning in section 4 is detailed in section 3. A computer scientist unfamiliar with quantum mechanics would be well served to reference chapter 2 of [NC00] in order to gain a deeper understanding of the notation covered in section 3 of this paper. Likewise, a quantum physicist would be well served to reference chapter 15 of [Garg04] to more deeply understand the fault modeling of distributed computing covered in section 2 and section 5.

2. DISTRIBUTED CONSENSUS

Consensus is a fundamental problem in distributed computing[Garg04]. Consider a distributed database in which a transaction spans multiple sites. For this transaction, it is important that either all sites agrees to commit or all sites agree to abort. In the absence of failures, this is a simple problem. We can use centralized or quorum-based algorithms to solve it. But what if processes fail? If links are reliable, surely the system can tolerate a single process failure? Surprisingly, it has been proven by Fischer, Lynch, and Paterson that consensus in the presence of even one unannounced process death is impossible to solve[FLP85].

This proof is widely accepted and is certainly valid in the scope of classical (non-quantum) communication but this paper intends to refute this proof in the more general case of mixed (quantum and non-quantum) communication. Since the physical reality of our universe is quantum mechanical and not classical, I would contend that the original FLP impossibility proof is therefore false in a general sense because it ignores the possibility of non-classical algorithms.

3. QUANTUM COMPUTING NOTATION

The algorithms in this paper depend on the mathematical framework of quantum mechanics. Although true quantum mechanics is a description of how real world physical systems isolated from their environment evolve over time, this paper will treat them in a completely mathematically abstract way. This allows for the construction of quantum algorithms without restricting the implementation to a specific physical realization. This generalization is both practical as well as necessary since the field of theoretical quantum computing is greatly ahead of practical physical realizations of such systems. In fact, current state of the art laboratory techniques are required just to successfully control even 5 qubits for under a second[Dalt05].

Despite this current limitation, there is much that can be learned by postulating how future systems will be able to use quantum resources. To achieve this goal, the following sections will make use of standard Dirac notation for describing quantum states. This means qubits will be represented as $|0\rangle$ and $|1\rangle$ as opposed to standard bits which are simply 0 and 1. The difference between the two being that a qubit can be in a linear combination of states called superpositions:

$$\psi = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

Where the coefficients α and β in equation 1 are complex numbers. By definition, the state of an unmeasured qubit is necessarily a two-dimensional complex vector space. This unitary vector space is often referred to as a Hilbert space. The Hilbert space is simply the union of all possible valid vector assignments for a given arrangement of qubits before measurement. Upon measurement, the quantum state collapses to one of the two basis states ($|0\rangle$ or $|1\rangle$) based on the probability implied by α and β. However, during the evolution of unmeasured quantum states, all transitions are unitary and linear to preserve a valid Hilbert space.

Tensor products (denoted by $\otimes$) are also used in the construction of this paper's algorithm. A tensor product creates a higher order vector space that combines multiple qubits' Hilbert spaces into a single, larger Hilbert space. In fact, the interaction between each individual qubit's Hilbert space with that of other qubits' Hilbert spaces is a critical feature of quantum mechanics upon which this research relies.

There are also references in this paper to quantum entanglement. When a qubit is entangled with one or more other qubits, its state becomes dependent not only on future operations performed on it but also on operations performed on qubits that it was entangled with. Although qubits are generally in direct physical proximity with each other when entanglement first occurs, they need not stay near each other to remain entangled. This phenomenon is the mechanism that permits non-local events to occur in quantum mechanics.

Section 5.4 also presents an example that refers to a Hadamard gate. For our purposes, a Hadamard gate is simply a mathematical construct (matrix) that maps a qubit in one basis state into an equally weighted superposition of both basis states. So passing $|0\rangle$ through a Hadamard gate transforms the qubit into equal parts $|0\rangle$ and $|1\rangle$. Passing that same qubit through another Hadamard gate would then put it into the $|1\rangle$ state.

4. QUANTUM DISTRIBUTED CONSENSUS ALGORITHM

The quantum mechanical resource that allows processes to achieve distributed consensus is known as the GHZ state[GHZ89]. The GHZ state is an ensemble of N quantum bits (qubits), represented by the entangled state

$$GHZ = \frac{|0\rangle^{\otimes N} + |1\rangle^{\otimes N}}{\sqrt{2}} \qquad (2)$$

Given this resource, we need only assume that each networked participant can receive a single qubit during the algorithm's setup phase and later measure it. With this lone assumption added to the classical information processing ability of the networked processors, consensus is achieved by the following algorithm:

1. Prepare the GHZ state by entangling a set of N qubits (see equation 2)
2. For N processes, distribute a single one of the N qubits from the superposition to each one of the participants in the system
3. Each participant measures their individual qubit
4. Chose 0 if the qubit measurement is $|0\rangle$ and chose 1 if the qubit measurement is $|1\rangle$.

This achieves distributed consensus for a single bit of information between multiple networked processes. Given that this algorithm is somewhat subtle and counter-intuitive, an elaboration of each step is provided.

1. The first step prepares a specially designed entangled state (GHZ state) whose non-local effects may be exploited later. This maximally entangled state provides deterministic, full correlation between measurements, even across the large physical distances present in typical distributed systems.
2. The second step accomplishes the distribution of the shared state between all participants. Even though each participant has only one qubit worth of quantum information, each one is in an entangled state. In other words, the state cannot be completely captured by the summation of the individual local states. Instead, the processes are now sharing highly correlated global state information.
3. The third step is where the uniquely quantum behavior of the algorithm is present. If this were not a quantum algorithm, this step would make the entire procedure trivial. There is no validity in a consensus algorithm simply distributing the same value to all participants and having them choose it. Entanglement prior to distribution is the key resource that makes it so that even though there is perfect symmetry between the qubits, none of their values have yet been decided.
4. Upon the first measurement by any participant, their qubit's state will collapse to either $|0\rangle$ or $|1\rangle$ with equal probability. Not only that, but every other participants' qubit will also instantaneously collapse to the exact same value as the first participant who measured it due to the laws of quantum mechanics.

This quantum distributed consensus algorithm provides all the required properties of standard distributed consensus: agreement, validity, and wait-free processing[Garg04]. Agreement is provided inherently by the laws of quantum mechanics. Theoretically, the measurement of any single fully-entangled qubit in the GHZ state will cause all other qubits in the same GHZ state to collapse to an identical ground state[NC00]. Indeed, this counter-intuitive theoretical correlation has been recently reconfirmed in an experiment involving three qubits[RWZ05]. Also, validity is provided when the first process that measures their qubit effectively proposes either $|0\rangle$ or $|1\rangle$. This decision is not predetermined before distributing the qubits, so this process is not equivalent to passing out a sealed envelope with the same value in it to all participants. In addition, wait-free processing is also inherently provided by the laws of physics, which instantaneously changes the state of all entangled qubits in the GHZ ensemble upon any single measurement.

5. FAULT MODELS

This algorithm is already known to work on reliable, synchronous networks with correct processes[DP06]. What is not known is the algorithm's resilience to different fault models.

5.1 ASYNCHRONOUS NETWORK DELAYS

Although originally proposed to run on a synchronous network[DP06], nothing about this algorithm requires synchronous messaging. Transmission only occurs from the initiator of the consensus to the other participants. Therefore, if there are no process faults or network faults, the operation of the algorithm is identical to a synchronous network. Delayed messages, though troublesome for classical algorithms, are no problem for this quantum variant. The other processes can continue executing even if a participant in the algorithm does not receive its qubit in a timely manner. In fact, when the delayed qubit finally arrives, it will already contain the agreed upon value. This late message will maintain the group consensus without undermining its validity.

5.2 PROCESS CRASH FAILURE SEMANTICS

Normally the use of an asynchronous network combined with the possibility of even a single crash is enough to undermine the ability to reach distributed consensus[FLP85]. This is because an arbitrarily slow response to a request cannot be differentiated from a process crash. Without being able to tell the difference, the negotiations required to reach a non-trivial consensus cannot occur. A traditional algorithm is forced to accept non-termination or non-validity to avoid the bivalent state problem inherent in the distributed consensus problem. Is the same true of this quantum consensus algorithm as well? Surprisingly, the answer is no. This algorithm requires only a one-way broadcast and no classical network reply. Assuming, of course, that the process preparing and distributing the original GHZ state in step 1 of the algorithm does not crash before completing step 2, the algorithm will complete for all other processes. The effect of any single process not measuring their qubit in step 3 will not impact the outcome of the computation, so the remaining correct processes can continue to operate in their absence.

5.3 BYZANTINE PROCESS FAILURE SEMANTICS

Using similar reasoning to that used in the previous section, it is possible to show that this algorithm is also able to operate in the face of malicious attempts to undermine it. No process is able to delay the results of the agreement protocol, so the wait-free nature cannot be defeated even in the face of an arbitrary number of byzantine processes. Validity cannot be undermined by a malicious process either. In fact, the laws of quantum mechanics precludes malicious manners in which a process could measure its own qubit. Any measurement will not affect the correlation of the remaining qubits, therefore an adversarial process that only has quantum measurement capabilities cannot tamper with the value the system ultimately chooses. It will either be $|0\rangle$ or $|1\rangle$ regardless of who measures it first or how that measurement is carried out. In addition, agreement is assured for all correct processes even if a majority of the processes willfully disregarded the correct bit value and chose the opposite value. This outcome is still superior to conventional algorithmic counter-measures to byzantine process failure because this algorithm cannot be put into a state where it cannot decide on either course of action the way other algorithms can[LSP82].

5.4 QUANTUM TAMPERING FAILURE SEMANTICS

Another scenario to consider is the consequences for the algorithm if adversarial networked processes are not only able to perform projective measurements, but also possess the power to perform single qubit operations on their own qubit. It has been experimentally shown that single qubit rotations are sufficient to reduce entanglement in the GHZ state[RRH+06]. Fortunately, the design of the distributed consensus algorithm prevents individual processes from thwarting the computation even in this way. Assume the first networked process has the first qubit in the shared GHZ state referenced in equation (2) and that it plans to tamper with its qubit by rotating it with a Hadamard gate,

$$H=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \tag{3}$$

Carrying out this rotation leads to,

$$H(1)[GHZ]=|1\rangle^{\otimes 1}+\frac{|0\rangle^{\otimes 2..N}+|1\rangle^{\otimes 2..N}}{\sqrt{2}} \tag{4}$$

This procedure leaves the untampered qubits completely entangled as does any single-qubit rotation. In fact, all procedures to deterministically reduce or eliminate entanglement between two or more qubits requires coordinated rotations by multiple processes. In addition, this collusion only affects the fidelity of their own coupling to the shared state. Since a correct process would not engage in malicious behavior to undermine the algorithm, correct processes will not be making rotations before their qubit measurements, so they will not be affected by this sort of tampering. As with classical byzantine processes, no procedure can force a truly independent process in the network to obey the consensus if it is determined not to agree.

6. LIMITATIONS

Given this new consensus algorithm, what limitations are there for its use in a distributed system? Although it has properties that no classical algorithm can have, it also has limitations that need to be considered.

6.1 PRACTICAL LIMITATIONS

Although based on the laws of physical reality as we currently know them, quantum communication systems have been notoriously difficult to build in practice[NC00]. Entangling a quantum state is difficult[HHR+05]. Advanced laboratory techniques currently allow only 5 qubits to be simultaneously entangled in the GHZ state used in the consensus algorithm presented here[Dalt05]. Additionally, normal laboratory conditions only allow this sort of entanglement to last on the order of 2 femtoseconds, although recent work has extended this time to over 24 hours under special conditions[HH04][Yang06].

Despite the difficulty of building a system that can control qubits, it is worth noting that occasional decoherence due to interference from the environment is not completely destructive to this algorithm. A spontaneous measurement of a qubit will only result in a premature measurement of the consensus bits before the algorithm completes. This will not result in a invalid algorithmic outcome. In this case, the processes will still all agree on the same value once the qubits are delivered even though none of the processes willfully made the proposal. It could be argued that this agreement is somewhat artificial since the outcome of $|0\rangle$ or $|1\rangle$ was determined by the decoherence of the system before the algorithm could take place. In one way this is a short-coming but it does also make the algorithm robust against some of the practical limitations of normal quantum computing. As soon as a physical system capable of carrying out this algorithm exists, it will not be nearly as sensitive to decoherence errors as other quantum algorithms like those for search and factoring[Grov96][Shor94] which require complete coherence across hundreds of qubits for several thousand quantum interactions.

6.2 THEORETICAL LIMITATIONS

The nature of the GHZ state that allows non-local effects to eliminate network delays that thwart classical distributed consensus algorithms is not without its drawbacks. As noted above, practical limitations in our ability to manipulate and control delicate quantum systems has not yet been realized outside of carefully controlled laboratories. Additionally, theoretical limitations exist due to the probabilistic nature of quantum wave-function collapse. Even though the system can deterministically agree upon a single bit in a bounded amount of time, the freedom to chose which value is ultimately proposed is limited. The first participant measuring the qubit has an equal chance of proposing 0 and an equal chance of proposing 1, but no straightforward way to propose either one by choice. This means that if the outcome of the distributed consensus was used to perform a task like determining whether to deduct money from a bank account, it would be a poor algorithm indeed. The bank would have a correctly operating system in the sense that all cash machines would agree about whether the money had been given out, but the man standing at the ATM will be disappointed 50% of the time when he's told that the quantum consensus engine has determined that he cannot have his money this time. There are however, other applications where this approach to consensus is still sufficient and useful. Consider a distributed linked list where two processes are inserting items at the same time. Inserting the two items in either order is equally correct, so in this scenario, the quantum algorithm yields the benefit of absolute distributed consensus without the probabilistic nature of the wave-function collapse limiting its usefulness.

7. IMPLICATIONS

This work shows that previous results[FLP85] about the impossibility of distributed consensus in the face of a single faulty process only takes advantage of a limited subset of physical resources available in our universe. Although modern processors and communication networks are not yet able to handle quantum information[NC00], the symmetry creating effects of the procedure presented in this paper are not simply a theoretical loophole. This algorithm provides a possible real-world way to escape the impossibility of distributed consensus in our physical reality.

Given this result, further study should be conducted on the enabling power of quantum resources in distributed computing. The non-local, synchronous behavior of quantum mechanics offers new and exciting possibilities to the field. Several previous results in the strictly classical computing environment have established bounds on what is thought possible. Perhaps quantum computing can extend some more of those bounds. For instance, what are the implications of quantum synchronizers given various network architectures? Can quantum teleportation solve data sharing problems resulting from a lack of shared-memory? What about mutual exclusion primitives? Can quantum locks extend the power of traditional locks, and if so, for which applications? Are there even possibilities more bold than quantum locks such as procedures to access the same variable simultaneously from two or more threads, thereby precluding the necessity of locks and critical sections altogether? There is currently very little known about the intersection of quantum and distributed computation. It seems unlikely that distributed consensus should be the only major result that needs to be revisited given the enabling power of quantum resources.

References

[Bell82] Bell, J. S. 1982. On the impossible pilot wave. Foundations of Physics, vol. 12, no. 10, 989-999.

[Dalt05] Dalton, B. J. 2005. Decoherence rates in large-scale quantum computers and macroscopic quantum systems. Journal of Modern Optics. 52(17):2563-2587.

[DP06] D'Hondt, E., Panangaden, P. 2006. The Computational Power of the W and GHZ states. quant-ph/0412177.

[FLP85] Fischer, M. J., Lynch, N. A., Paterson, M. S. 1985. Impossibility of distributed consensus with one faulty process. Journal of the ACM, 32(2), 374-382.

[Garg04] Garg, V. 2004. Concurrent and Distributed Computing in Java. John Wiley & Sons.

[GHZ89] Greenberger, D. M., Horne, M. A., Zeilinger, A. 1989. Bell's theorem, Quantum Theory, and Conceptions of the Universe (edited M. Kafatos), 73-76. Dordrecht: Kluwer Academic Press.

[Grov96] Grover, L. K. 1996. A fast quantum mechanical algorithm for database search. Proceedings of the 28th Annual ACM Symposium on Theory of Computing.

[HHR+05] Häffner, H., Hänsel, W., Roos, C. F., et. al. 2005. Scalable multiparticle entanglement of trapped ions. Nature 438, 643-646.

[HH04] Hughes, R., Heinrichs, T. 2004. A Quantum Information Science and Technology Roadmap. http://qist.lanl.gov

[Land88] Landauer, R. 1988. Dissipation and noise immunity in computation and communication. Nature 335. 779-784.

[LSP82] Lamport, L., Shostak, R., Pease, M. 1982. The Byzantine Generals Problem. ACM Transactions on Programming Languages and Systems. 4 (3): 382-401.

[NC00] Nielsen, M., Chuang, I. 2000. Quantum computation and quantum information. Cambridge University Press.

[RRH+04] Roos, C. F., Riebe, M., Häffner, H., Hänsel, W., Benhelm, J., Lancaster, G., Becher, C., Schmidt-Kaler, F., Blatt, R. 2004. Control and Measurement of Three-Qubit Entangled States. Science 304(5676), 1478-1480.

[RWZ05] Resch, K. J., Walther, P., Zeilinger, A. 2005. Full characterization of a three-photon GHZ state using quantum state tomography. Phys. Rev. Lett. 94, 070402. quant-ph/0412151v1

[Shor94] Shor, P. W. 1994. Algorithms for quantum computation: discrete logarithms and factoring. Proceedings, 35th Annual Symposium on Fundamentals of Computer Science (FOCS). p.124- 134.

[Yang06] Yang, T., et al. 2006. Experimental Synchronization of Independent Entangled Photon Sources. Physical Review Letters 96, 110501.