Plug-in quantum software

John Preskill

Some quantum states are hard to create and maintain, but are a valuable resource for computing. Twenty-first century entrepreneurs could make a fortune selling disposable quantum states.

In principle, quantum computers could solve problems that conventional digital computers could never solve[1]. In practice, quantum hardware is in its infancy, and has far to go before quantum computers can realize their promise. Perhaps someday, as suggested on page 390 of this issue by Daniel Gottesman of Microsoft Research and Isaac Chuang of the IBM Almaden Research Center[2], the capabilities of quantum computers will be significantly extended by 'quantum software' that can be prepared offline and shipped to users over the 'quantum Internet'.

Whereas conventional computers process information encoded in bits, a quantum computer processes information encoded in quantum states, such as the internal electronic states of individual atoms, or the spin states of atomic nuclei. Existing quantum computers contain just a few atoms, and can perform only very simple computations. But because the inherent complexity of a quantum state rises very steeply as the number of atoms increases, a quantum computer acting on thousands of atoms could have staggering power. For example, today's digital supercomputers would take billions of years to find the prime factors of a number that is hundreds of digits long, whereas a quantum computer of the future might perform that task in seconds. Constructing large-scale quantum computers will be a formidable technological challenge, so it is not yet possible to predict when this revolutionary technology might become reality.

A quantum software program is a particular quantum state that enables a quantum computer to perform a specific task. If that quantum state is difficult or inconvenient to prepare, the user of a quantum computer might prefer to acquire the state from a vendor, rather than prepare it herself. The user's hardware then acts on the quantum state according to a standard protocol, but the outcome varies depending on the version of the software. Unfortunately for the user (but to the delight of the vendor) quantum software would be a consumable product, unavoidably damaged after a single use (Fig. 1). Thus we can foresee the flourishing of a quantum software industry. A manufacturer can design a valuable quantum state, and use a special-purpose device to churn out multiple copies of the state; these can be tested to assure quality and stored until needed. Consumers can download the state for a fee and plug it into their own quantum computers to achieve improved performance.

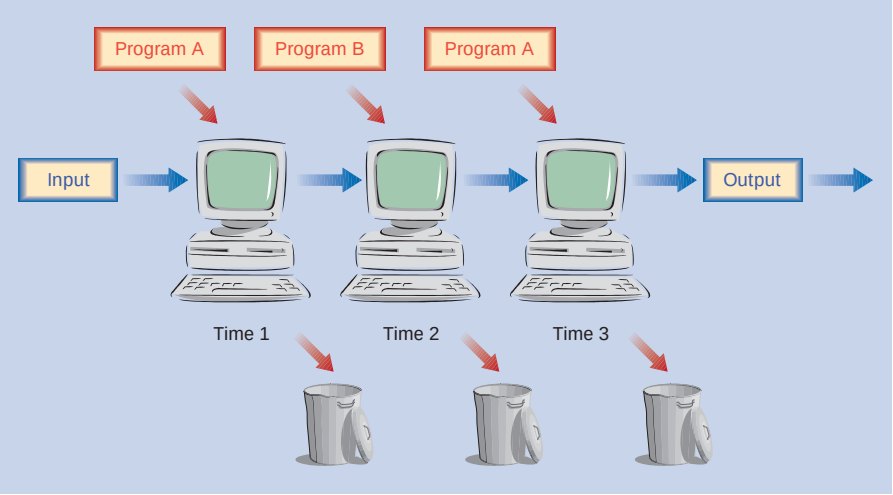

Figure 1 **Disposable quantum software. Gottesman and Chuang[2] suggest that perhaps, one day, quantum software might be delivered from vendor to user over a quantum communication network (the 'quantum Internet'). After a single use, the software is irreparably damaged and must be discarded. To execute a quantum algorithm, the user might have to download and consume a particular program many times.**

One important application for quantum software will be to ensure that quantum computers function reliably. A quantum computer is equipped to execute a standard set of unitary transformations, called its quantum gates. By executing many gates in succession, a computer applies a complex unitary transformation to an input quantum state, yielding an output state that is ultimately measured. The set of standard gates may be finite, but a well-chosen gate set is universal, enabling the computer to simulate any specified unitary transformation to any desired accuracy[3].

Because complex quantum states are extraordinarily fragile, the quantum-computing hardware that implements the gates needs to meet very demanding specifications. But even with superb hardware, a computer could achieve acceptable reliability only by applying the recently discovered principles of quantum-error correction: if quantum information is cleverly encoded, it can be protected both from the destructive effects of uncontrolled interactions with the environment, and from the cumulative effect of the inevitable small imperfections in the hardware[4,5]. Furthermore, for a given coding scheme, particular quantum gates that are well matched to the structure of the code can be executed fault tolerantly — that is, even though the hardware that implements the gate is imperfect, the action on the encoded quantum information is highly accurate. Fault-tolerant gates that form a universal set allow a quantum computation to be executed with good reliability.

For each of the known quantum-error correction schemes, some of the gates in the fault-tolerant universal set are easy to perform, whereas others are hard. These 'hard' gates would be most conveniently executed with quantum software that can be prepared ahead of time and then consumed during the operation of the gate. Realizing the gate with software rather than hardware is preferable because we could verify before use that the software has been prepared according to specifications, and we can discard or repair it if it is found to be faulty. In contrast, if our hardware fails badly during the execution of a quantum gate, it might be difficult to recover from the damage.

The idea of preparing special states offline to allow fault-tolerant quantum computation is not new[6,7]. But Gottesman and Chuang[2] now propose a novel and more systematic approach to the preparation, use and classification of these states. Their central idea is that applying a quantum software routine can be viewed as a generalized form of

quantum teleportation. In standard teleportation[8] one party (Alice) destroys an unknown quantum state, and then sends a classical message to another party (Bob), who proceeds to reconstruct a perfect replica of the original state. To achieve this feat, the two parties consume some standard quantum software that they share. Gottesman and Chuang note that if the software is suitably modified, the reconstructed quantum state is modified too. Instead of winding up with Alice's original state, Bob reconstructs a state to which a quantum gate has been applied. If Alice and Bob use different software, they execute a different gate. Standard teleportation has been demonstrated in the laboratory[9–11], and some of the protocols suggested by Gottesman and Chuang are feasible, or nearly so, with existing techniques.

A quantum computer with sufficiently sophisticated hardware could run on classical software. But a more affordable machine might be marketed with only rudimentary hardware tools. Unable to execute a complete universal set of fault-tolerant gates with its hardware alone, it would achieve that capability through quantum software suited to its coding scheme. Users of such computers would require broadband access to the quantum Internet[12], over which reliable quantum software could be shipped on demand.

Sometime during the twenty-first century (no one can say just when), fault-tolerant quantum computers made possible by quantum software may achieve processing speeds far surpassing those of conventional digital computers. But before that happens, quantum software could already be a marketable commodity. Because tampering with a quantum state leaves a detectable imprint, a shared quantum state can enable two parties to establish a secure communication link[13,14]. To exploit the growing demand for privacy, a quantum software company might create a product that allows two customers to talk to one another without fear of eavesdropping, or to authenticate each other's identities. Although quantum computers will be needed to create and distribute the software, these devices would be less complex and easier to build than quantum computers that solve complex computational problems.

Another intriguing development is the recent discovery[15] by Daniel Jonathan and Martin Plenio of Imperial College, London, that in a particular setting quantum software can serve as a catalyst for a quantum operation — it enables execution of the operation without being consumed in the process. Will quantum states be found that can catalyse fault-tolerant gates or other useful operations? The invention of reusable quantum software would be deflating news for the stockholders of 'Quantum Valley', but would be welcomed by tomorrow's consumers of quantum information technology. ■

John Preskill is in the Division of Physics, Mathematics, and Astronomy, California Institute of Technology, Pasadena, California 91125, USA.
e-mail: preskill@theory.caltech.edu

1. Shor, P. in *Proc. 35th Annu. Symp. on Foundations of Computer Science* (ed. Goldwasser, S.) 124–134 (IEEE Press, Los Alamitos, California, 1994).
2. Gottesman, D. & Chuang, I. L. *Nature* **402,** 390–393 (1999).
3. Deutsch, D. *Proc. R. Soc. Lond. A* **425,** 73–90 (1989).
4. Shor, P. *Phys. Rev. A* **52,** R2493–R2496 (1995).
5. Steane, A. M. *Phys. Rev. Lett.* **77,** 793–797 (1996).
6. Shor, P. in *Proc. 35th Annu. Symp. on Foundations of Computer Science* (ed. Goldwasser, S.) 56–65 (IEEE Press, Los Alamitos, California, 1994).
7. Knill, E., Laflamme, R. & Zurek, W. H. *Proc. R. Soc. Lond. A* **454,** 365–384 (1998).
8. Bennett, C. H. *et al. Phys. Rev. Lett.* **70,** 1895–1899 (1993).
9. Bouwmeester, D. *et al. Nature* **390,** 575–579 (1997).
10. Boschi, D. *et al. Phys. Rev. Lett.* **80,** 1121–1125 (1998).
11. Furusawa, A. *et al. Science* **282,** 706–709 (1998).
12. Cirac, J. I., Zoller, P., Kimble, H. J. & Mabuchi, H. *Phys. Rev. Lett.* **78,** 3221–3224 (1997).
13. Bennett, C. H. & Brassard, G. in *Proc. IEEE Int. Conf. Comp. Sys. Sig. Process.* 175 (IEEE Press, New York, 1984).
14. Ekert, A. K. *Phys. Rev. Lett.* **67,** 661–663 (1991).
15. Jonathan, D. & Plenio, M. B. *Phys. Rev. Lett.* **83,** 3566–3569 (1999).

Botany

The family tree flowers

Paul Kenrick

The origin and relationships of flowering plants are among evolutionary biology's enduring mysteries. These comparative newcomers to the evolutionary stage number a staggering 250,000 living species classified into about 350 families. One of the cherished goals of botany is to unravel the complicated family tree, a process that involves establishing the branching relationships among species (phylogeny). Two papers in this issue take an impressive step forward. Soltis *et al.* (page 402)[1] and Qiu *et al.* (page 404)[2] use a multigene approach to identify the closest living relatives of flowering plants and map out the deepest branches of the family tree.

Analysing the deep phylogenetic relationships in flowering plants is a complicated business. This is partly because of the sheer size and diversity of the group, and partly because of its rapid diversification during the Early Cretaceous (130–90 million years ago)[3]. Flowering plants differ considerably from their closest living relatives in the gymnosperms (conifers and their allies), creating additional problems for 'rooting' the family tree. Recent molecular systematic work — which includes some of the largest phylogenetic analyses attempted for any group of organisms — confirms the existence of a major group (eudicots) which has a characteristic, three-aperture pollen type and includes most of the dicotyledons[4,5]. The widely recognized division between

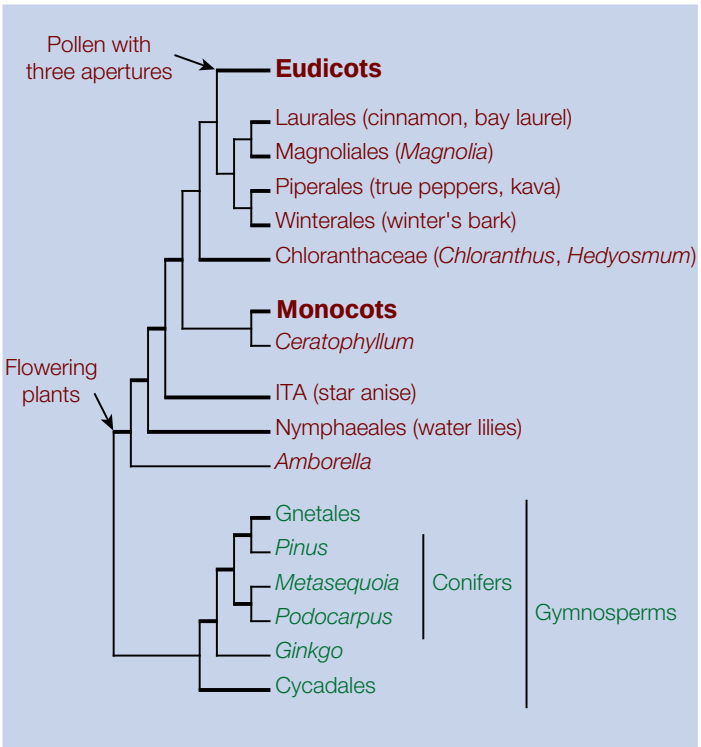

Figure 1 **Summary of relationships among the major groups of flowering plants and gymnosperms based on the multigene sequence analysis of Qiu *et al.*[2]. Thicker branches are those with strongest (90–100% bootstrap) support from the analysis. In this scheme, dicotyledons are not recognized as a group within the flowering plants.**