Re: qubits

Source: http://coding.derkeiler.com/Archive/General/comp.object/2004−10/0593.html

---

From: H. S. Lahman (*h.lahman_at_verizon.net*)
Date: 10/12/04

Date: Tue, 12 Oct 2004 14:02:06 GMT

Responding to Martin...

> *Simultaneously running a set of decoherent states through the same*
> *logic is an interesting idea. Only those initial states that could*
> *make it all the way through would pop out the other end.*
> *Hypothetically you could assemble every trip of the traveling*
> *salesman as a set of simultaneous states and run them through a*
> *minimization algorithm. One cycle of the algorithm would collapse*
> *the state space to the minimum solution, taking an NP−complete*
> *problem and making it linear. Or so my meager recollection of the*
> *theory goes.*

I agree its an interesting idea, but color me skeptical. Often the problem lies in setting up the problem properly. For example, to "assemble every trip of the traveling salesman problem" might be an O(N**2) operation in itself. More to the point, I think one needs to separate the problem formulation and solution algorithm from the hardware implementation. Most np−Complete problems have been mathematically proven to be such _for a particular problem formulation_.
That is intrinsic in the problem statement and the np−Completeness does not depend upon the computational mechanisms. Thus any Qubit improvement for existing np−Complete problem formulations would be brute force rather than introducing an inherent algorithmic improvement like collapsing a state space. [Some np−Complete problems have been solved in O(NlogN) time by reformulating the problem into a tractable dual. Revised Dual Simplex is based, in part, on this idea.]

Though Qubits has obvious potential for increasing computing power at the hardware level, I suspect we would need some rather drastic advances in software technology before making use of it. Fundamentally almost all of today's software is about digital computing; expressing solutions to problems in terms of binary states. Currently we can't even optimize 3GLs for multiple processors at the macro level, much less deal with n−ary states at the "bit" level.

We would need a completely different computational model for the hardware and dealing with that would be like starting software engineering all over again using plug boards. For example, today's 3GLs

would likely be useless because even the notion of stack−based architecture represents serialization that may be irrelevant in a Qubit computational model, much less dealing with a multi−valued notion of Truth in conditions. Our experience today might allow us to travel the road faster, but it will be just as long.

However, I think the problem goes further than software. Digital computing is so ubiquitous today that entire mathematical disciplines like numeric methods are dependent on it (e.g., analysis of precision). That spreads out tendrils by influencing the way people design algorithms. That, in turn, brings us full circle to my point above: the way we /define/ problems today is subtlely affected by the digital computing paradigm (albeit quite indirectly). So taking advantage of Qubits is going to require a lot of brainwashing even outside the software community before we could begin to use them effectively. On the upside, the post−brainwashing reformulation of problems could make the solutions more tractable as dual formulations. B−)

****
There is nothing wrong with me that could
not be cured by a capful of Drano.

H. S. Lahman
hsl@pathfindermda.com
Pathfinder Solutions −− Put MDA to Work
http://www.pathfindermda.com
blog (under constr): http://pathfinderpeople.blogs.com/hslahman
(888)−OOA−PATH