

Improving Quantum Computer Simulations

Quantum computers would exploit the unusual behavior of the smallest particles of matter and light. Their theoretical ability to perform vast numbers of operations simultaneously has the potential to solve certain problems, such as breaking data encryption codes or searching large databases, far faster than conventional computers [1]. Promising candidates for use as quantum bits (qubits) in quantum computers are ions (electrically charged atoms), the nuclear spins of certain atoms in specially designed molecules (addressed by nuclear magnetic resonance techniques) and electron spins trapped in quantum dots (serving as artificial atoms).

Despite of the impressive development during the last years, an experimental demonstration that quantum computation can solve a non-trivial problem is still lacking [2]. To be of practical use, quantum computers will need error correction, which requires at least several tens of qubits and the ability to perform hundreds of gate operations. A physically realizable quantum computer is a complex many-body quantum system. In order to exercise control over many qubits and to suppress the rate at which errors are introduced during a quantum computation (decoherence and systematic errors due to imperfections of pulse sequences), it is indispensable to understand the time-dependent behavior of the whole system. In addition, the ability to simulate realistic models of quantum computers that interact with their environment is crucial for the successful realization of scalable quantum computing hardware – which is

currently the most significant barrier to building a working quantum computer.

A starting point are simulations at gate level. Each quantum operation is represented by a quantum gate which acts instantaneously on the state vector of the qubits. Since the dimension of the state vector grows exponentially with the number of qubits, the simulation of quantum computers is clearly memory bounded. For example, 3TB of memory are required to simulate 37 qubits. Highly optimized memory access and communication patterns are needed for efficient simulation. For this purpose the “Massively Parallel Quantum Computer Simulator” described in [3] has been developed. This code exhibits very good scaling as a function of the number of processors.

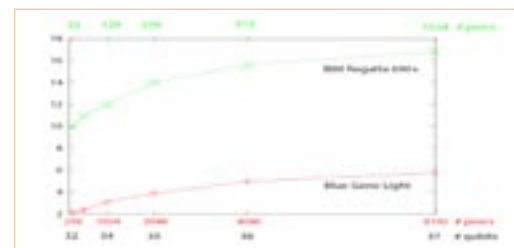


Figure 1: Scaling (with fixed local system size) on IBM Regatta 690+ and Blue Gene/L using up to 1024 (8192) processors respectively. The time per quantum operation is given as a function of the global system size

Since the number of operations per CPU is constant – adding one qubit doubles the size of the state vector and therefore the number of CPUs – ideal scaling would correspond to a line parallel to the x-axis. Indeed, for large systems one finds such an asymptotic behavior.

In principle, because of its “universality”, this gate level code can also be used



Figure 2: Quantum Fourier Transformation: the ensemble average spin vector and the cloud of measured endpoints at the end of each single experimental run. The leftmost qubit is $q=1$

to simulate physical systems, such as quantum spin models or models for physical realizations of quantum computers, by writing the time evolution of the physical system as a sequence of elementary quantum gate operations [1]. However, this approach is bound to be computationally inefficient for all but nontrivial physical systems. Instead, it is much more effective to allow for additional unitary transformations that implement operations such as the time evolution under a Heisenberg Hamiltonian in an optimal manner. This extension does not affect the intrinsic performance of the code.

Within the framework of gate level simulations, it is nevertheless possible to implement a basic error model which covers a) operational errors and b) decoherence [4, 5]. The underlying idea is that every single qubit operation can be generated from plane rotations and phase shifts. This decomposition allows introducing angle and phasing errors. Computation of controlled multi-qubit gates can be reduced to effective single qubit operations that act only on the part of the state vector whose control-bit(s) are set to $|1\rangle$. To study decoherence effects, a bit-flip, a phase-flip or both are introduced with probability $p/3$ each. The state vector remains unchanged with probability $1-p$.

Applying this error model to the Quantum Fourier Transformation and Grover's quantum search algorithm, one finds that the QFT circuit is more robust to operational inaccuracies than Grover's algorithm on comparable scales. Critical

parameters can be derived which give a first estimate of tolerable error thresholds. These results will be compared with future calculations from dynamic simulations of quantum computer devices, taking into account the full time evolution according to a time dependent Hamiltonian describing both, the system and the environment.

References

We thank the DEISA Consortium (co-funded by the EU, FP6 project 508830), for support within the DEISA Extreme Computing Initiative (www.deisa.org).

- [1] Nielsen, M.A., Chuang, I.L.
Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, 2000
- [2] Di Vincenzo, D.P.
<http://arxiv.org/abs/quant-ph/0002077>
- [3] De Raedt, K., Michielsen, K., De Raedt, H., Trieu, B., Arnold, G., Richter, M., Lippert, Th., Watanabe, H., Ito, N.
Massively Parallel Quantum Computer Simulator, to be published in Computer Physics Communication, 2006
- [4] Niwa, J., Matsumoto, K., Imai, H.
<http://arxiv.org/abs/quant-ph/0201042>
- [5] Arnold, G., Richter, M., Trieu, B., Lippert, Th.
Improving Quantum Computer Simulations, to appear in the proceedings of the AQIS06 conference

- Guido Arnold
- Marcus Richter
- Binh Trieu

John von Neumann
Institut für
Computing (NIC)
Forschungszentrum
Jülich

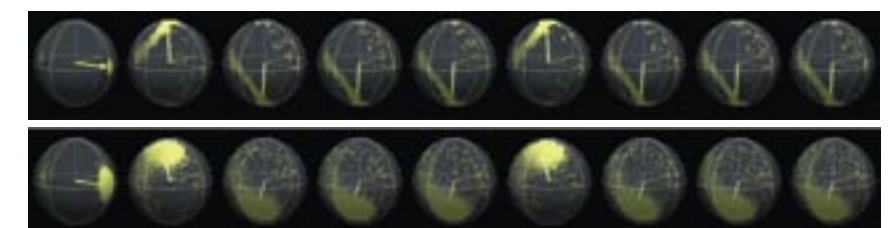


Figure 3: Grover's quantum search algorithm: the 8 qubits encode the searched data base element $k=17=10001$ (the leftmost qubit is ancillary). The upper line corresponds to a subcritical error distribution while the parameter of the lower sequence is above the threshold