文章编号: 1001-2486 (2000) 02-0086-04

# 量子算法模拟实现技术研究*

谭明锋, 戴 葵, 刘 芸

(国防科技大学计算机学院, 湖南 长沙 410073)

摘 要: 量子计算并行性展示了强大的运算能力, 但现实条件决定了目前量子算法研究主要依靠模拟进行。本文介绍了量子计算的基本原理和计算模型, 对量子算法的模拟实现技术进行了分析。通过对量子算法模拟的时空分析, 指出目前量子算法模拟中存在的一些问题, 讨论了今后的一些研究方向。

关键词: 量子算法; 模拟; 实现技术

中国分类号: TP 391.9; 0413.1 文献标识码: A

## Techniques for Simulating Quantum Algorithms

TAN Ming-feng, DAI Kui, LIU Yun

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract**: The parallelism of quantum computing has shown its great power. However, the reality of the quantum computation enforces the researchers to simulate the quantum algorithms instead of doing research work on the real quantum computers. Based on the elementary theory and the simple model of quantum computation simulation, this paper discusses the techniques of quantum algorithm simulation. And according to the analysis of the space and time consuming of these simulation, we point out some problems in quantum algorithm simulation. we also discussed the techniques for optimizing the quantum algorithm simulation in the future.

**Key words**: quantum computation; simulation; technique of simulation.

1994 年, Peter Shor 提出了分解大数质因子的量子算法, 将大数质因子分解的时间复杂度, 由普遍认为的冯. 诺依曼计算机上的指数级降低为在量子计算机上的多项式级[1]; 另外, Grover 提出的对非结构化数据库的搜索量子算法可获得相对经典算法 $O(\sqrt{N})$ 的加速比。这些量子算法都利用了量子效应极大地加速了普通计算。但是到目前为止, 量子计算机还没有走出实验室的大门, 而对于量子算法在冯. 诺依曼计算机上的模拟, 可以为研究者们提供一个虚拟的量子计算机平台, 作为进一步进行研究的重要工具和手段。量子算法模拟实现的研究对量子计算理论、算法研究和量子算法可行性、正确性研究都具有重要意义。

本文中首先简要介绍一些对量子算法模拟较重要的量子力学基本概念和理论, 然后分析实现量子算法模拟的关键技术, 并讨论量子算法模拟实现中存在的问题和解决这些问题的途径。

## 1 量子算法模拟实现技术

### 1.1 量子计算原理

在量子计算机中, 数据通过量子寄存器中所有量子位的共同量子状态来表示。一个 $n$ 位普通寄存器只处于唯一的状态中, 而由量子力学的基本假设[2], 一个 $n$ 位量子寄存器可处于 $2^n$ 个基态的相干叠加态 $|\psi\rangle$中, 即可同时表示 $2^n$ 个数。叠加态和基态的关系可以用狄拉克记号表示为:

$$|\psi\rangle = \sum_i c_i |\phi\rangle \tag{1}$$

式中 $c_i$ 为复数, 称为概率幅, $|c_i|^2$ 给出了 $\psi$ 在受到与量子计算机系统相纠缠的测量仪器观测而发

* 收稿日期: 1999-09-27
基金项目: 国家部委预研基金
作者简介: 谭明锋(1976-), 男, 计算机学院硕士生。

生脱散时塌缩到基态 $|\phi\rangle$的概率, 即对应得到结果为数 $i$ 的概率, 因此有 $\sum_i |c_i|^2 = 1$。

量子寄存器中的量子态通过量子门的作用演化为另一个量子态, 量子门的作用与逻辑电路门类似。在指定基态的条件下, 量子门可以由作用于希尔伯特空间中向量的矩阵 $\hat{A}$ 描述。由于量子门的线性约束, 量子门对希尔伯特空间中量子状态的作用将同时作用于所有基态上, 对应到 $n$ 位量子计算机模型中, 相当于同时对 $2^n$ 个数进行运算。这就是量子并行性[3]。在Shor 算法算法中利用了这一点。

量子计算的一个主要原理是: 使构成叠加态的各个基态通过量子门的作用互相干涉, 从而改变它们之间的相对相位, 使概率幅发生变化, 从而使各个由基态所表示的运算结果被观测到的概率发生变化。如一个叠加态为 $|\psi\rangle = \frac{2}{\sqrt{5}}|0\rangle + \frac{1}{\sqrt{5}}|1\rangle = \frac{1}{\sqrt{5}}\begin{pmatrix}2\\1\end{pmatrix}$, 设一个作用于其上的量子门为 $\hat{O} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$, 则二者的作用结果是 $|\psi'\rangle = \frac{3}{\sqrt{10}}|0\rangle + \frac{1}{\sqrt{10}}|1\rangle$。可以看到, 基态 $|0\rangle$ 的概率幅增大, 而 $|1\rangle$ 的概率幅减小。Grover 等量子算法主要利用了这一量子机制。另一个重要的量子计算机制是量子纠缠态, 它违背我们的直觉。对处于纠缠态的量子位中的某几位进行操作, 不仅会改变这些量子位的状态, 而且会改变与它们相纠缠的其它量子位的状态[3]。在 Shor 算法中得到量子傅立叶变换所需要的状态时, 就用到了这一经量子力学特性。

**1.2** 量子计算模型和量子算法模拟的计算机表示

量子计算是量子系统初态 $|\psi_0\rangle$经过与控制系统的相互作用, 随时间演化为所需末态 $\psi_m$ 的过程。量子计算的控制系统由量子门 $U_i(i=1\cdots m)$、在 $U_i$ 后的测量算子 $O_i$ 表征, 计算的中间结果由态矢序列 $\{\psi_1\psi_1\cdots\psi_{m-1}\ \psi_{m-1}\}$ 表征。因此, 量子计算的一个简单模型可以定义如下:

$$\begin{aligned}&\hat{O}_m\hat{U}_m\cdots\hat{O}_1\hat{U}_1(|\psi_0\rangle)\\&= \hat{O}_m\hat{U}_m\cdots\hat{U}_2\hat{O}_1(|\psi_1'\rangle) = \hat{O}_m\hat{U}_m\cdots O_2U_2(|\psi_1\rangle)\\&= \hat{O}_m(|\psi_m'\rangle) = |\psi_m\rangle\end{aligned} \quad (2)$$

研究量子计算时, 通常简化了量子状态。由式(1), 在选定基态后, 叠加态的信息可用基态的概率幅 $c_i$ 及对应基态的编号表示。对 $n$ 位量子寄存器而言, 其中的叠加态经过变换得到的新状态, 仍然保存在这个寄存器中。因而量子寄存器的表示必须要有能力保存所有 $2^n$ 个基态的信息。所以在实现时, $n$ 位量子寄存器可以由一个表示所有基态的数据结构和寄存器的附加信息表征, 无须专门表示叠加态。而作用于 $n$ 位量子寄存器的量子门可由一个 $2^n\times 2^n$ 的酉矩阵表示, 相应数据结构用 $2^n\times 2^n$ 二维数组保存。

**1.3** 量子算法模拟实现分析

模拟运算的控制流程由所模拟的计算或算法决定, 其中可能有诸如循环、条件判断、调用等等, 主要用于避免不必要的空间消耗。所有这些控制手段, 在一般的高级语言中都有提供, 在此不再更多叙述。

与 3.1.1 中量子计算模型对应, 模拟实现过程可分为以下步骤:

(1) 定义量子寄存器和量子门, 其中包括位数 $n$、所有量子门的矩阵元素 $u_{i,j}$ 等;

(2) 初始化量子态, 即将每个基态的概率幅设置为 $1/2^{\frac{n}{2}}$;

(3) 根据算法控制一序列量子门矩阵与寄存器中表示叠加态的数组做幺正变换, 模拟并行计算函数结果或改变特定基态的概率幅;

(4) 根据算法对某些量子位进行测量;

(5) 判断是否结束。结束则终止。未结束则重复 3、4。

考虑式(2)中的一个步骤 $U(\psi_k) = \psi_{k+1}$, 最直接的方法是用如(3)矩阵乘法实现, 模拟程序段为:

```
for (k = 0; k < 2^n; k++)   C[i] = u[i][k] * a[k];
```

$$\begin{pmatrix}u_{0,0} & \cdots & u_{0,2^n-1}\\ & \ddots & \\ u_{2^n-1,0} & \cdots & u_{2^n-1,2^n-1}\end{pmatrix}\left(c_0\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix} + c_1\begin{pmatrix}0\\1\\\vdots\\0\end{pmatrix} + \cdots + c_{2^n-1}\begin{pmatrix}0\\0\\\vdots\\1\end{pmatrix}\right) = \begin{pmatrix}u_{0,0} & \cdots & u_{0,2^n-1}\\ & \ddots & \\ u_{2^n-1,0} & \cdots & u_{2^n-1,2^n-1}\end{pmatrix}\begin{pmatrix}c_0\\c_1\\\vdots\\c_{2^n-1}\end{pmatrix} \quad (3)$$

然而可以看到, 这种方法需要大量的空间来模拟量子门。Barenco 等人的研究结果表明, 完备的量子门集合应包括: $C^{not}$门、所有的一位量子门, 以及相位移动变换$\begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}$, $\begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix}$。也就是说, 一个 $2^n$ 维的量子门可以分解为 $O(2^n)$ 个 2 维或 4 维的这些量子门的直积和复合[4]。通过这种分解, 可以将模拟量子门所需要的空间由 $O(2^{2^n})$ 减少为 $O(2^n)$, 而且在模拟时, 并不需要再合成为一个大的矩阵、再进运算, 而只需按顺序将分解后的低维量子门分别作用于相应的量子位即可。这样, 不但没有增加时间开销, 而且还由原来的 $O(2^{3n})$ 减少为 $O(2^{2n})$。

模拟时每个基态的概 '率幅已知, 其模的平方即为应测得该基态的概率。设基态数为 $N$, 测量的实现即要求设计一算法, 使选出这 $N$ 个基态中的每一个的频率在总次数趋于无穷时逼近其实际测得的概率。设 $N$ 个基态对应概率为 $P_1 \cdots P_N$, 在数轴上表示如下, 每个区间的长度表示概率值的大小:

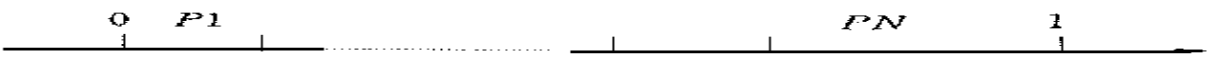


图 1

Fig. 1

随机选择 $a$, 使 $0 \le a \le 1$, 则 $a$ 落在最大概率确定的区间中的概率也最大。若 $P_{i-1} < a < P_i$, 则选出第 $i$ 个基态。由概率论中的大数定理知, 设算法中得到第 $i$ 个基态次数为 $f$, 总次数为 $F$, 有: $\lim_{F \to +\infty} f / F = P_i$。这样就找到了满足要求的算法。

## 2 进一步分析

从前面的分析可看到, 在没有任何模拟优化时, 对 $n$ 位 $m$ 个门的量子进行模拟对主机存储容量要求至少是 $C = 4m2^{2n} + 2^{n+3}$ 字节。假设单机磁盘容量为 10GB, $m = 8$, 则它能够模拟表示 14 位的量子寄存器。量子算法模拟过程可看作是矩阵相乘的过程, 模拟过程中所要完成的乘加操作至少有 $m \times 2 \times 2^{2n}$ (FLOPS)。假设单机的运算速度为 1000MFLOPS, 那么运算时间 $T$ (单位为小时) 和量子位数 $n$ 之间的关系为 $T = m \times 2 \times 2^{2n} / (1000M \times 3600)$ (h)。那么对 14 个量子位规模的系统进行模拟大约需要 58min。但这一估算并没有考虑磁盘访问时间, 而且容易算出, 14 位的量子寄存器最大只能表示 16383。我们用 Java 语言实际模拟了 Shor 算法和 Grover 算法的执行。首先, 建立了一系列类库, 用于表示概率幅、量子状态、量子变换、函数定义等等。通过调用这些类库, 可以模拟量子寄存器、量子门、测量, 从而实现了量子计算的执行。在这个量子计算模拟系统基础上, 模拟了 Shor 算法的计算过程。算法的并行部分直接采用用串行模拟实现, 通过这种未加优化的模型, 可以作为我们进一步工作的原型和比较对象。图 2 和图 3 是部分对 Shor 算法模拟的结果, 其中的输出寄存器指的是 Shor 算法中求出指数模后保存函数结果的寄存器。

从图中可以看出, Shor 算法失效的出现还是比较频繁的, 而且甚至会出现完全失效的情况( 例如无法用此算法分解 27)。这是由于这里分解的都是较小的数, 同时 Shor 算法的物理基础是非确定性的量子计算机的缘故。还可以看到, 由于质数本身不可分解, 因此量子算法模拟存在判断何时停机的问题。但也正是由于这种不确定性, 使量子计算机具有空前的运算能力。

量子算法模拟所需要的时间、空间与模拟规模成指数关系, 由前面的分析可以看到, 空间的瓶颈要比时间瓶颈严重。为了解决这些阻碍量子算法模拟的问题, 必须对量子算法模拟技术进行优化, 发掘量子算法模拟的潜力。例如: 采用特殊的表示方法和计算方法减小时空开销; 针对特定算法或算法中的特定步骤的优化技术以提高时空效率; 用分布式模拟技术和多机系统并行模拟技术将空间或时间开销分布到多个子系统中以提高模拟速度和规模等等。这些都可能是发展量子模拟实现技术的突破口。我们正在进一步进行对量子算法模拟进行优化的技术和量子算法的研究。

## 3 结论

量子算法模拟在目前和将来较长一段时间里, 将是研究、验证、发展量子计算的理论、算法和机制的

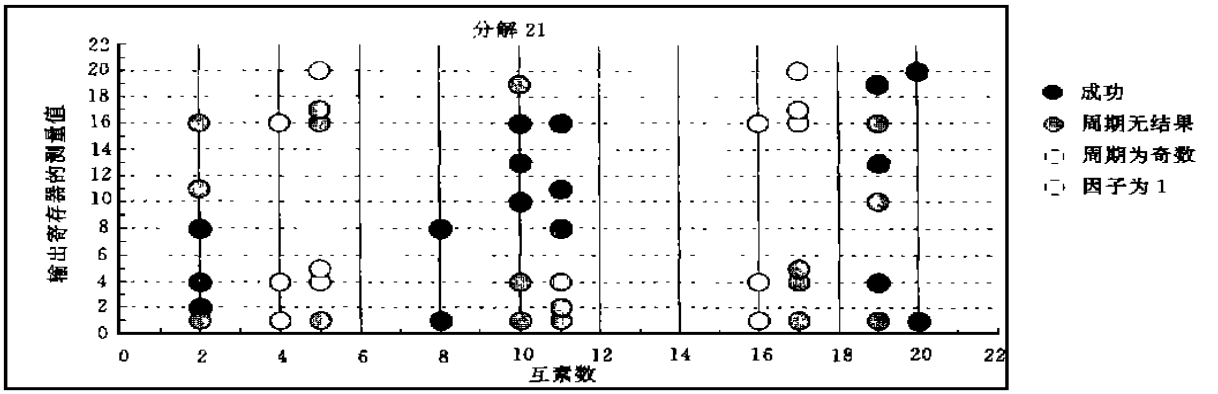


图 2 Shor 算法分析 21

Fig. 2 Factoring 21 with Shor´s Algorithm

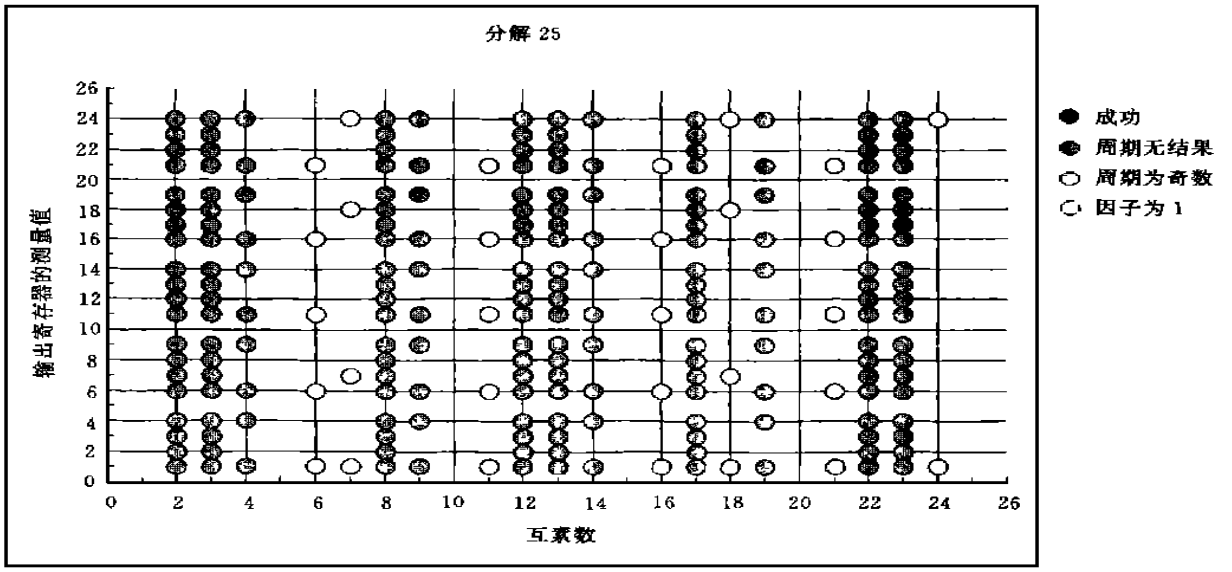


图 3 Shor 算法分解 25

Fig. 3 Factoring 25 with Shor´s Algorithm

重要手段。文中着重分析了量子算法模拟实现的关键技术,并讨论了由量子计算特点所决定的时间、空间耗费巨大的问题。可以看到,目前量子算法模拟只能模拟有限规模的量子计算。研究如何才能在现有条件下更好的模拟量子算法问题是一个紧迫的任务,这也是我们的一个研究方向。

## 参考文献:

[1] Shor Peter. Polynomial_ Time Algorithms for Prime Factorization and Discrete Logarithms on Quantum Computr [J]. SIAM Journal of Computing, 26(5): 1484～1590.

[2] 邹鹏程. 量子力学(第一版) [M], 北京: 高等教育出版社, 1989.

[3] Valerio Scarani. Quantum Computing [R]. Institu de Physique Exp rimentale, Ecolc Polytechnique F d rale de lausannne, CH-1015 Lausanne, Switzerland, email: valerio. scarani @ epfl. ch. November. 1998.

[4] Robert R. Ticci A Rudimentary Quantum Compiler [C]. P. O. Box 266, Bedford, MA 01730 November 17, 1998, email: tucci@ar-tiste. com.

[5] Bernhard Omer. A Procedural Formalism for Quantum Computing [EP/OL]. DepartMent of Theoretical Physics, Techenical University of Vienna, 23 th July 1998 email: oemer @ tph. tuwien. ac. at.

[6] Richard Jozsa. Qutantum Effects in Algorithms [EB/OL]. School of Mathematics and Statistics, University of Plymouth, Plymouth, Devon PL48AA, U. K, email: rjozsa@ plymouth. ac. uk.