Quantum Applications

Dr. Sebastian Feld

Von Schaltkreisen zu Quantum Services

Vielleicht hat alles im Mai 1981 begonnen, als Richard Feynman in einem Plenarvortrag die Frage gestellt hat, ob Quantenphysik effizient von klassischen Computern simuliert werden kann. Die Antwort lautet schlichtweg: nein. Es musste also eine andere Lösung her. In kurzer Folge wurden entsprechende Formalisierungen vorgestellt, nämlich das quantenmechanische Modell einer Turingmaschine (Paul Benioff, 1982) als Analogon einer klassischen Rechenmaschine oder sogenannte Quantengatter (David Deutsch, 1985), das Pendant zur klassischen Binärlö- gik. Im Grunde war somit ein theoretischer Quantencomputer konzipiert. Es stellte sich nun jedoch eine weitere Frage: Mit welchen Anweisungen kann dieser theoretische Quantencom- puter welche Aufgaben schneller oder besser als ein klassischer Computer lösen? Es dauerte einige Jahre, bis Algorithmen gefunden wurden, die bewiesenermaßen schneller als ihre entsprechenden klassischen Verfahren sind. So hat Lov Grover 1996 einen Weg gefunden, um eine unsortierte Datenbank schnell zu durchsuchen, und Peter Shor gelang es 1997 ein Verfahren zu beschreiben, mit welchem eine zusammengesetzte Zahl effizient in ihre Teiler zerlegt werden kann. Von diesen theoretischen Errungenschaften getrieben kam auch der expe- rimentelle Drang auf zu überprüfen, ob solch eine Maschine tatsächlich gebaut und besagte Algorithmen praktisch ausge-

führt werden können. Und es war tatsächlich möglich: 2001 haben experimentelle Physiker, Ingenieure und viele weitere kluge Köpfe eine Maschine gebaut, die in der Lage war, Shors Algorithmus auszuführen. Die Wissenschaftler haben die Zahl 15 in ihre Primfaktoren 3 und 5 zerlegt.

Diese scheinbar einfache Rechnung ($3 \times 5 = 15$) war wissen- schaftlich gesehen jedoch ein unbeschreiblich großer Schritt. Hiermit wurde bewiesen, dass Shors theoretische Versprechung auch praktisch umgesetzt werden kann. Denn mal ehrlich: Was nutzt einem ein theoretischer Bauplan einer Zeitmaschine, wenn diese nicht praktisch gebaut werden kann. Es folgten zahlreiche Durchbrüche in Sachen Ingenieurskunst, die der allgemeinen Öffentlichkeit eher verborgen geblieben sind. Der Bau eines funktionierenden Quantencomputers verlangt enormes theoretisches und praktisches Können aus vielen unterschiedlichen Disziplinen. Und genau dieses Wissen und diese Erfahrungen wurden in den darauffolgenden Jahren an- gehäuft und erprobt, sodass wir in den letzten etwa 5 Jahren zahlreiche Erfolge im Bereich der Hardware-Entwicklung beobachten konnten. Anscheinend haben die Ergebnisse die Labore der Universitäten verlassen und Einzug in die F&E-Ab- teilungen der globalen Unternehmen genommen. Verschiedene Akteure aus Wirtschaft und Wissenschaft forschen nun an sehr unterschiedlichen technischen Kandidaten zur Realisie-

rung von Qubits, dem essentiellen Grundbaustein eines jeden Quantencomputers. Und all diese Kandidaten besitzen ihre eigenen Vorteile, Unzulänglichkeiten und entsprechend auch „Technology Readiness Levels“.

Nachdem nun erfolgreich daran gearbeitet wird, die Hardware (d.h. den Quantencomputer) aus der Theorie in die Praxis zu bringen, muss dies ebenso auch für die Software (d.h. die Algorithmen) geschehen. Oder anders ausgedrückt: Wir besitzen eine ausgereifte Idee von Quantencomputing, bewiesene theoretische Formalisierungen und erste praktische Realisierungen der Technik, und nun gilt es, die „PS auf die Straße zu bringen“ und Quantencomputer tatsächlich praktisch einzusetzen. Dies fühlt sich allerdings wie eine Art Zeitreise an. Es existiert Erfahrung aus Jahrzehnten professioneller Softwareentwicklung mit Design Patterns, Best Practices und dergleichen. Es offenbart sich jedoch, dass wir vieles des bereits Bekannten erneut erlernen müssen. Mit Sicherheit werden wir einige Techniken und Ideen der klassischen Softwareentwicklung auch im Quantencomputing verwenden können (bspw. Code-Reusability), anderes scheint jedoch viel schwerer zu sein (bspw. Fehlerkorrektur) oder gar nicht möglich (Inspektion von Variablen). Sprich: Wir müssen lernen, neu zu denken.

Wenn wir nun eine Zeitreise getan haben, wo befinden wir uns dann? Aktuell scheinen wir das Ende der Phase „Kernel-Entwicklung“ zu erreichen. Bislang haben wir uns mit einer 1-zu-1-Beziehung von Lösung zu Problem befasst, es wurden starre Schaltkreise entwickelt, die genau ein spezifiziertes Problem lösen. Diese Vorgehensweise ist vergleichbar mit der Programmierung mittels Lochkarten bis in die 1960er-Jahre hinein. Die nächste Phase, die nun zaghaft beschritten wird, ist wohl die „Algorithmus-Entwicklung“. Im Fokus stehen also nicht mehr kleine Mengen von Quantengattern, die als Einheit (Kernel) angesehen werden, sondern vielmehr Module. Es geht vorrangig um eine Wiederverwendbarkeit von Code, welche jedoch immer noch stark an der zugrunde liegenden Technik angelehnt ist und entsprechendes Wissen um die Quantenphysik erfordert. Aus Sicht der Software ist schließlich das Ziel, die Phase der „Modell-Entwicklung“ zu erreichen. Diese ähnelt der vorherigen Phase, ist jedoch weiter abstrahiert: Der Fokus entfernt sich gänzlich von Details der Quantenphysik und

beinhaltet nun weitaus stärker notwendiges domänenspezifisches Wissen (bspw. aus den Bereichen Chemie, Finanzen oder Logistik) mit dem Ziel, mächtige Bibliotheken anzubieten.

Auf die eben beschriebenen Phasen aufbauend wird nun die Software- und Informatik-Sicht etwas verlassen und eine eher wirtschaftliche Betonung gesetzt. So werden zukünftig die Phasen der „Quantum Applications“ und „Quantum Services“ folgen. Es sollen Anwendungen erschaffen werden, die einen messbaren Wert schöpfen und darüber hinaus mit den Begebenheiten der Unternehmen skalieren. Die Wertschöpfung kann einerseits innerhalb des Unternehmens stattfinden (Quantum Applications im Sinne einer Prozessoptimierung) und andererseits außerhalb des Unternehmens (Quantum Service als Dienstleistung und Wirtschaftsgut). Je weiter wir uns allerdings von den technischen Details entfernen, desto öfter scheint Verwirrung in der Diskussion des Für und Wider von Quantencomputing aufzutauchen. Es gilt wie so oft: Nicht jedes Werkzeug passt zu jeder Fragestellung. Grundsätzlich existieren leichte und schwere Problemstellungen, und zwar sowohl für klassische Computer als auch für Quantencomputer. Es ist jedoch nicht der Fall, dass ein Problem, das für klassische Computer schwer ist, automatisch einfach für einen Quantencomputer ist. Domänenexperten aus Wirtschaft und Wissenschaft sollten also frühzeitig mit einer sorgfältigen Analyse der „Kosten und Nutzen“ beginnen. Damit ist folgende Frage gemeint: Gegeben ist die ganz spezielle Problemdomäne eines Unternehmens. Welchen Business Value würde die Problemlösung bieten (Nutzen) und wo liegt der erwartete Zeitpunkt eines entsprechenden Quantenvorteils (Kosten)? Sie erinnern sich: Ein vorteilhafter Algorithmus, der zwar in der Theorie existiert (siehe Primfaktorzerlegung), muss erst noch praktisch realisiert werden können ($15=3 \times 5$). Leider ist die Beantwortung der soeben gestellten Frage, nämlich welches Problem wann mit welchem Vorteil gelöst werden kann, schwierig. Aber: Sie lohnt sich!

Lassen Sie uns in diesem Heft also eben dieser Frage nachgehen und schauen, wie wir vorteilbringende Quantum Applications entwickeln können und vielleicht auch wann wir diese nutzen können. Ich wünsche Ihnen viel Spaß bei der Lektüre!